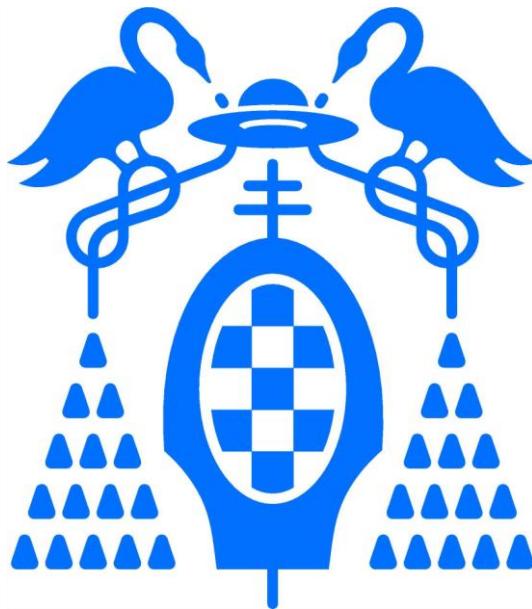


**UNIVERSIDAD DE ALCALÁ**

**Escuela Politécnica Superior**

**INGENIERIA TÉCNICA INDUSTRIAL**  
**ESPECIALIDAD EN ELECTRÓNICA INDUSTRIAL**



**Trabajo Fin de Carrera**

**ROBOT EUROBOT 2015**

**SISTEMAS DE MANIPULACIÓN DE ELEMENTOS DE JUEGO**

**D. Mario Inglés Garcés**

**SEPTIEMBRE 2015**

**UNIVERSIDAD DE ALCALÁ**

Escuela Politécnica Superior

**INGENIERÍA TÉCNICA INDUSTRIAL  
ESPECIALIDAD EN ELECTRÓNICA INDUSTRIAL**

Trabajo Fin de Carrera

**“ROBOT EUROBOT 2015, SISTEMAS DE  
MANIPULACIÓN DE ELEMENTOS DE JUEGO”**

**TRIBUNAL:**

Presidente: D. Pedro Alfonso Revenga de Toro

Vocal 1º: D Juan Carlos García García

Vocal 2º: D. Julio Pastor Mendoza

CALIFICACIÓN: .....

FECHA: .....



## *Agradecimientos y dedicatoria*

Son tantas las personas a las que tengo que agradecer su ayuda y apoyo para la realización de este trabajo, que creo que sería una lista demasiado larga y quizás injusta para alguna que pudiera quedar fuera, cada pequeña ayuda ha sido decisiva. Cada una de ellas ya sabe en todo lo que me ha ayudado para poder concluir este proyecto y cerrar una etapa. Tan solo quiero que sepan, que espero poder estar siempre ahí, cuando ellas me necesiten, aunque en ocasiones solo sea como apoyo moral.



# ÍNDICE GENERAL

---

ÍNDICE DE FIGURAS .....	IX
ÍNDICE DE TABLAS.....	XIII
ÍNDICE DE DIAGRAMAS .....	XV
ÍNDICE PLANOS.....	XVII
RESUMEN.....	XXI
SUMMARY .....	XXIII
PALABRAS CLAVE.....	25
RESUMEN EXTENDIDO .....	27
1. INTRODUCCIÓN Y OBJETIVOS .....	1
1.1. INTRODUCCIÓN .....	1
1.2. EUROBOT .....	1
1.2.1. <i>El concurso</i> .....	1
1.2.2. <i>La asociación de Eurobot</i> .....	3
1.3. ORIGEN DEL PROYECTO.....	4
1.4. PROBLEMA A RESOLVER Y OBJETIVOS .....	7
1.5. ESTRUCTURA DE LA MEMORIA .....	9
2. DISEÑO MECÁNICO .....	11
2.1. ESPECIFICACIONES DE DISEÑO.....	11
2.2. FASES DE DISEÑO MECÁNICO.....	15
2.3. ANALISIS Y DIMENSIONADO DE SISTEMAS .....	17
3. ESTRUCTURA PORTANTE .....	23
3.1. CUESTIONES DE DISEÑO PREVIAS .....	23
3.2. ESTRUCTURA PRINCIPAL Y SISTEMA DE ENSAMBLADO POR PESTAÑAS .....	23
3.3. SISTEMA DE TRACCIÓN.....	27
3.4. SISTEMA DE POSICIONAMIENTO ODOMÉTRICO.....	35
4. SISTEMA DE MANIPULACIÓN DE STANDS.....	39
4.1. PROTOTIPOS Y ESTUDIO PREVIO .....	41
4.2. SISTEMA DE PALETAS DE STANDS.....	46
4.3. SISTEMA DE ELEVACIÓN DE STANDS.....	49
4.3.1. <i>Carro pinza</i> .....	59
4.3.2. <i>Carro motor</i> .....	63
4.4. SISTEMA CARRO HORIZONTAL DE STANDS.....	64
4.5. SISTEMA DE PUERTAS DE STANDS.....	67
4.6. SECUENCIA DE CONSTRUCCIÓN DE FOCO.....	69
4.7. TIEMPO MÍNIMO DE PROCESADO DE STANDS .....	74
5. SISTEMA DE MANIPULACIÓN DE PALOMITAS .....	75

5.1.	PROTOTIPOS Y ESTUDIO PREVIO .....	78
5.2.	SISTEMA MÁQUINAS DE PALOMITAS .....	78
5.2.1.	<i>Rampas de palomitas</i> .....	82
5.2.2.	<i>Bandeja de palomitas</i> .....	83
5.2.3.	<i>Puertas de palomitas</i> .....	85
5.3.	PINZA VASO TRASERO DE PALOMITAS .....	87
5.4.	SISTEMA PINZA VASO DELANTERO DE PALOMITAS .....	89
6.	SISTEMA DE CIERRE DE CLAQUETAS .....	93
7.	ANÁLISIS DE DISEÑO GLOBAL .....	95
7.1.	ESTIMACIÓN DE MASA Y POSICIÓN DE CENTRO DE MASAS .....	95
7.2.	LISTADO PIEZAS MECANIZABLES .....	103
7.3.	LISTADO TORNILLERÍA .....	110
8.	ELECTRÓNICA DE CONTROL .....	113
8.1.	ARQUITECTURA HARDWARE .....	113
8.2.	TARJETA DE CONTROL .....	115
8.3.	SENSOR E3Z-D61 DE OMRON .....	116
8.4.	SERVOS FUTABA S3003 .....	117
8.5.	SERVOS DYNAMIXEL AX12 .....	118
8.6.	CONJUNTO MOTOR Y ENCODER MAXON .....	120
9.	SOFTWARE DE CONTROL .....	123
9.1.	ANALISIS PREVIO .....	123
9.2.	ARQUITECTURA SOFTWARE .....	124
9.3.	LECTURA Y PROCESAMIENTO DE SENsoRES .....	125
9.4.	CONTROL DE MECANISMOS .....	126
9.4.1.	<i>Interfaz y funciones de control de mecanismos</i> .....	126
9.4.2.	<i>Mecanismos basados en servos analógicos</i> .....	127
9.4.3.	<i>Mecanismos basados en servos AX12</i> .....	127
9.4.4.	<i>Intercambiador de stand</i> .....	128
9.5.	GESTIÓN DE MECANISMOS Y SISTEMAS MECÁNICOS .....	130
9.6.	INTERFAZ DE CONTROL I2C .....	131
	PLANOS .....	133
	PLIEGO DE CONDICIONES .....	385
1.	PLIEGO DE CONDICIONES DE DISEÑO .....	386
1.1.	CONDICIONES DE DISEÑO MECÁNICO .....	386
1.2.	CONDICIONES DE DISEÑO SOFTWARE .....	387
1.3.	CONDICIONES DE SEGURIDAD DEL DISEÑO .....	387
1.4.	CONDICIONES DE MATERIALES .....	387
2.	PLIEGO DE CONDICIONES TÉCNICAS DE FABRICACIÓN .....	389
2.1.	CONDICIONES GENERALES A EMPRESAS SUBCONTRATADAS .....	389
2.2.	CONDICIONES PARA EL MONTAJE .....	390
2.3.	PRUEBAS DE FUNCIONAMIENTO .....	390

PRESUPUESTO .....	391
1. INTRODUCCIÓN .....	392
2. PRESUPUESTO DE EJECUCIÓN MATERIAL .....	392
2.1. COSTE DE MATERIAL .....	393
2.2. COSTE DE MANO DE OBRA .....	400
2.3. COSTE DE AMORTIZACIÓN DE EQUIPOS Y SOFTWARE.....	401
2.4. TOTAL PRESUPUESTO DE EJECUCIÓN MATERIAL .....	401
3. GASTOS GENERALES Y BENEFICIO INDUSTRIAL .....	402
4. COSTE DE EJECUCIÓN POR CONTRATA .....	402
5. IMPORTE TOTAL DE PROYECTO.....	402
MANUAL DE USUARIO .....	403
1. PROTOCOLO DE COMUNICACIÓN I2C ENTRE MICROCOTROLADORES .....	404
1.1. ESTRUCTURA I2C_CMD_SLAVE_DSPIC_SET_MODE .....	404
1.1.1. <Comandos de inicialización> .....	405
1.1.2. <Comandos para control de actuadores simples> .....	405
1.1.3. <Comandos para control de actuadores múltiples>.....	410
1.1.4. <Comandos para ajuste de variables de estado> .....	412
1.2. ESTRUCTURA I2C_REQ_SLAVE_DSPIC_STATUS .....	413
ANEXOS.....	415
1. LIBRERÍAS AVERSIVE .....	417
1.1. ORGANIZACIÓN DE MÓDULOS .....	417
1.2. BASE/SCHEDULER .....	418
1.3. BASE/TIME .....	420
1.4. COMM/UART .....	422
1.5. COMM/I2C_SLAVE .....	423
1.6. DEBUG/ERROR .....	425
1.7. DEVICES/CONTROL_SYSTEM/CONTROL_SYSTEM_MANAGER .....	426
1.8. DEVICES/CONTROL_SYSTEM/FILTERS/PID .....	430
1.9. DEVICES/CONTROL_SYSTEM/FILTERS/QUADRAMP .....	434
1.10. DEVICES/ROBOT/BLOCKING_DETECTION_MANAGER .....	437
1.11. DEVICES/ENCODERS_DSPIC.....	440
1.12. DEVICES/SERVO/AX12 .....	441
1.12.1. Protocolo de comunicación y funciones de acceso a registros.....	441
1.12.2. Inicialización y configuración de la interfaz HW .....	445
1.12.3. Funciones de alto nivel .....	446
1.13. HARDWARE/DSPIC/PWM_SERVO .....	447
1.14. HARDWARE/DSPIC/PWM_MC.....	450
2. CONEXIONADO HARDWARE DETALLADO .....	453
2.1. CONEXIONADO DE PUERTOS EN DSPIC.....	453
2.2. MAPEO DE ENTRADAS-SALIDAS .....	456

3.	MANUAL DE IMPLEMENTACIÓN SW DETALLADO .....	457
3.1.	ARQUITECTURA SW DETALLADA .....	457
3.2.	CONTROL SERVOS AX12 .....	458
3.3.	COMANDOS UART PC DE DEPURACIÓN Y CONFIGURACIÓN.....	458
3.4.	ESTRUCTURA DE DATOS .....	460
3.5.	ESTRUCTURA DE PROGRAMA.....	462
3.5.1.	<i>Archivo “main.c”</i> .....	464
3.5.2.	<i>Archivo “state.c”</i> .....	464
3.5.3.	<i>Estructura funciones de ejecución de comandos de actuadores simples</i> .....	467
3.5.4.	<i>Estructura funciones de gestión de sistemas de actuadores múltiples</i> .....	469
3.5.5.	<i>Estructura funciones de ejecución de comandos de actuadores multiples</i> .....	471
3.5.6.	<i>Archivo “actuator.c”</i> .....	473
3.5.7.	<i>Estructura funciones de control de actuadores</i> .....	473
3.6.	LINEAS DE EJECUCIÓN .....	475
3.7.	GESTIÓN DE SCHEDULER. TAREAS DE EJECUCIÓN PERIÓDICA .....	478
3.8.	CONFIGURACIÓN MÓDULOS AVERSIVE.....	479
3.8.1.	<i>Scheduler_config.h</i> .....	479
3.8.2.	<i>Time_config.h</i> .....	479
3.8.3.	<i>Uart_config.h</i> .....	480
3.8.4.	<i>Pwm_servo_config.h</i> .....	480
3.9.	CODIFICACIÓN COMANDOS SLAVEDSPIC UART PC.....	480
3.10.	GESTIÓN DE EVENTOS I2C DE COMANDOS MAESTRO .....	482
3.10.1.	<i>i2c_write_event()</i> .....	483
3.10.2.	<i>i2c_read_event()</i> .....	484
3.11.	FUNCIONES DE GESTIÓN DE NUEVO COMANDO .....	484
3.11.1.	<i>i2c_set_mode()</i> .....	485
3.12.	SECUENCIA DE RESET POR LINEA DE COMANDOS “P O P” .....	486
3.13.	FUNCIÓN DE GESTIÓN DE REGISTRO DE ERRORES “MYLOG( )” .....	487
3.14.	CONFIGURACIÓN Y GESTIÓN DE MÓDULO CONTROL DE BUCLE CS PARA SISTEMA MOTOR-ENCODER ..	488
3.15.	FUNCIÓN DE GESTIÓN DE MÓDULOS “DO_CS( )” .....	488
3.16.	ESTRUCTURA DE CONTROL DE BUCLE CS .....	489
3.16.1.	<i>Función de configuración de módulo cs “slavedspic_cs_init( )”</i> .....	490
3.16.1.1.	Calculo de coeficiente de velocidad y aceleración.....	494
3.16.1.2.	Configuración filtro de consigna “Quadramp” .....	495
3.16.1.3.	Configuración y ajuste filtro de corrección “PID”.....	496
3.16.1.4.	Configuración y ajuste de módulo de detección de bloqueo .....	496
3.17.	FUNCIONES DE NIVEL INTERMEDIO PARA CONTROL DE ACTUADORES “ACTUATOR.C” .....	499
3.17.1.	<i>Funciones de control de servos AX12</i> .....	499
3.17.2.	<i>Funciones de control del carro horizontal</i> .....	502
3.17.3.	<i>Funcion de control de bandeja de palomitas</i> .....	505
3.17.4.	<i>Función de control de pinzas de elevadores de stands</i> .....	505
3.17.5.	<i>Funciones de control de puertas de liberación de torres de stands</i> .....	505
4.	ESQUEMAS ELÉCTRICOS .....	509
	BIBLIOGRAFIA .....	511

# ÍNDICE DE FIGURAS

FIGURA 1-1 CAMPO EUROBOT 2015, CON ELEMENTOS DE JUEGO .....	27
FIGURA 1-2 ROBOT "PETINTO" VISTA DELANTERA DISEÑO 3D A LA IZQUIERDA, FOTOGRAFÍA A LA DERECHA ...	28
FIGURA 1-3 ARQUITECTURA HARDWARE. DIAGRAMA DE BLOQUES.....	31
FIGURA 1-4 DIAGRAMA DE CAPAS PRINCIPAL DE LA ARQUITECTURA SOFTWARE.....	32
FIGURA 1-1 ROBOT "PETINTO" VISTA DELANTERA DISEÑO 3D A LA IZQUIERDA, FOTOGRAFÍA A LA DERECHA .....	4
FIGURA 1-2 ROBOT "PETINTO" VISTA TRASERA DISEÑO 3D A LA IZQUIERDA, FOTOGRAFÍA A LA DERECHA .....	5
FIGURA 1-3 CAMPO EUROBOT 2015, CON ELEMENTOS DE JUEGO (EUROBOT, 2014) .....	6
FIGURA 2-1 SECUENCIA DE MOVIMIENTOS DE ESTRATEGIA OPTIMIZADA SOBRE IMAGEN (EUROBOT, 2014) ....	12
FIGURA 2-2 CONJUNTO BLOQUES VISTA DELANTERA.....	18
FIGURA 2-3 SECUENCIA 1-3.....	19
FIGURA 2-4 CONJUNTO BLOQUES VISTA TRASERA.....	21
FIGURA 3-1 PERSPECTIVA ESTRUCTURA EXPLOSION .....	24
FIGURA 3-2 PERSPECTIVA ESTRUCTURA IZQUIERDA DELANTERA, DERECHA TRASERA.....	25
FIGURA 3-3 PERSPECTIVA ESTRUCTURA INFERIOR .....	26
FIGURA 3-4 DISPOSICIÓN SISTEMA DE LASTRE EN BASE ROBOT EN TORNO A BLOQUE MOTOR.....	26
FIGURA 3-5 VISTA PERSPECTIVA BLOQUE MOTOR.....	27
FIGURA 3-6 VISTA LATERAL BLOQUE MOTOR.....	27
FIGURA 3-7 GRAFICA SELECCIÓN CORREAS DENTADAS .....	29
FIGURA 3-8 CALCULO DESARROLLO POLEA (ZAYAS FIGUERAS, 2008) .....	30
FIGURA 3-9 CURVA MOTOR A ALIMENTACIÓN NOMINAL (HUBRICH, 2006).....	31
FIGURA 3-10 CONJUNTO BALANCÍN .....	36
FIGURA 3-11 BRAZO HORIZONTAL 90°.....	37
FIGURA 3-12 BRAZO GIRO 1° .....	37
FIGURA 3-13 PERSPECTIVA CONJUNTO BISAGRA Y BALANCÍN ENCODER, A LA IZQUIERDA PERSPECTIVA Y A LA DERECHA VISTA LATERAL .....	37
FIGURA 3-14 SISTEMA ENCODER EN ROBOT VISTA LATERAL .....	38
FIGURA 3-15 ABERTURA ENCODER EN BASE ROBOT .....	38
FIGURA 4-1 SISTEMAS DE MANIPULACIÓN DE STANDS .....	39
FIGURA 4-2 DISEÑO DE ALTERNATIVA DE ELEVACIÓN POR CORREA .....	43
FIGURA 4-3 DETALLE DE SISTEMA DE PALETAS EN BASE DE ROBOT CON BLOQUEO DE STAND .....	46
FIGURA 4-4 DETALLE CENTRADO STAND SISTEMA DE PALETAS .....	46
FIGURA 4-5 BLOQUE DE SISTEMA DE PALETAS.....	47
FIGURA 4-6 VISTA EN PLANTA DE SISTEMA DE PALETAS CON POSIBLE DISPOSICIÓN 1 .....	48
FIGURA 4-7 VISTA EN PLANTA DE SISTEMA DE PALETAS CON POSIBLE DISPOSICIÓN 2 .....	48
FIGURA 4-8 VISTA FRONTAL DE SISTEMA DE PALETAS DE STANDS .....	48
FIGURA 4-9 SISTEMA ELEVACIÓN DERECHO EN POSICIÓN "ABAJO" Y SISTEMA DE PALETAS.....	49
FIGURA 4-10 SISTEMA ELEVACIÓN DERECHO EN POSICIÓN "ARRIBA" Y SISTEMA DE PALETAS.....	49
FIGURA 4-11 ESQUEMATIZACIÓN DE SISTEMA DE DOBLE MANIVELA-BIELA CON MEDIDAS DE DISEÑO DEFINITIVO.....	50
FIGURA 4-12 ESQUEMATIZACIÓN DE SISTEMA DE DOBLE MANIVELA-BIELA CON MEDIDAS PARA SIMPLIFICACIÓN DE ECUACIÓN DE MOVIMIENTO EN EL EJE VERTICAL .....	51
FIGURA 4-13 RESOLUCIÓN GRÁFICA DEL MOMENTO RESISTENTE MÁXIMO CON RADIO DE MOMENTO EN VALOR MÁXIMO .....	52
FIGURA 4-14 RESOLUCIÓN GRÁFICA DEL MOMENTO RESISTENTE MÁXIMO CON FUERZA SOBRE MANIVELA EN VALOR MÁXIMO .....	52

FIGURA 4-15 SISTEMA IMPLEMENTADO PARA ESTUDIO DE SISTEMA DE ELEVACIÓN EN MS ADAMS VISTA 3D .....	54
FIGURA 4-16 SISTEMA IMPLEMENTADO PARA ESTUDIO DE SISTEMA DE ELEVACIÓN EN MS ADAMS CON REPRESENTACIÓN DE TODOS LOS ELEMENTOS INCLUIDOS (ARTICULACIONES, GUÍAS, ETC.) .....	54
FIGURA 4-17 GRAFICA MS ADAMS CINEMÁTICA CARRO PINZA EN EJE Y .....	55
FIGURA 4-18 MOMENTO MÁXIMO SOBRE MANIVELAS Y COORDENADA Y DE EJE MANIVELA-BIELA INFERIOR.....	55
FIGURA 4-19 MOMENTO MÁXIMO SOBRE MANIVELAS CON ROZAMIENTO DINÁMICO EN GUÍAS DE 0.05. ....	56
FIGURA 4-20 GRAFICA CINEMÁTICA CARRO PINZA EN EJE Y PAR MOTOR EN FUNCIÓN DE RAD/S .....	57
FIGURA 4-21 SECUENCIA ELEVACIÓN.....	58
FIGURA 4-22 VISTA DELANTERA DE CARRO PINZA IZQUIERDO .....	59
FIGURA 4-23 VISTA TRASERA DE CARRO PINZA IZQUIERDO .....	59
FIGURA 4-24 DETALLE DISPOSICIÓN DE ELEMENTOS EN DISEÑO DE CARRO PINZA .....	60
FIGURA 4-25 SISTEMA PINZA EN POSICIÓN DE APERTURA MÁXIMA (GIRO 45°) .....	61
FIGURA 4-26 SISTEMA PINZA EN POSICIÓN DE APERTURA MEDIA (GIRO 22.5°) .....	61
FIGURA 4-27 SISTEMA PINZA EN POSICIÓN CERRADA (GIRO 0°).....	62
FIGURA 4-28 DETALLE CONSTRUCTIVO CARRO PINZA IZQUIERDO EXPLOSIÓN .....	62
FIGURA 4-29 VISTA DELANTERA DE CARRO MOTOR IZQUIERDO .....	63
FIGURA 4-30 VISTA LATERAL DE CARRO MOTOR IZQUIERDO .....	63
FIGURA 4-31 DETALLE ENSAMBLADO DE CARRO MOTOR POR PESTAÑAS .....	64
FIGURA 4-32 CONJUNTO DE SISTEMA DE MANIVELAS POR PLATO DE SERVO .....	64
FIGURA 4-33 CONJUNTO ELEMENTOS CARRO HORIZONTAL SOBRE BASE ROBOT .....	64
FIGURA 4-34 CARRO HORIZONTAL EN MITAD DE RECORRIDO .....	66
FIGURA 4-35 CARRO HORIZONTAL EN MITAD DE RECORRIDO VISTA LATERAL .....	66
FIGURA 4-36 DETALLE ELEMENTOS SISTEMA CARRO HORIZONTAL EXPLOSIONADOS .....	66
FIGURA 4-37 SISTEMA DE PUERTAS DE STANDS INTEGRADO EN SISTEMA DE MANIPULACIÓN DE STANDS .....	67
FIGURA 4-38 SISTEMA PUERTAS DE STANDS EN POSICIÓN CERRADAS CON DETALLE DE CORDONES ELÁSTICOS DE CIERRE.....	68
FIGURA 4-39 SISTEMA PUERTAS DE STANDS CON PUERTA DERECHA ABIERTA. .....	68
FIGURA 4-40 SECUENCIA FOCO 1.....	69
FIGURA 4-41 SECUENCIA FOCO 2.....	69
FIGURA 4-42 SECUENCIA FOCO3.....	69
FIGURA 4-43 SECUENCIA FOCO 4.....	69
FIGURA 4-44 SECUENCIA FOCO 5.....	70
FIGURA 4-45 SECUENCIA FOCO 6.....	70
FIGURA 4-46 SECUENCIA FOCO 7.....	70
FIGURA 4-47 SECUENCIA FOCO 8.....	70
FIGURA 4-48 SECUENCIA FOCO 9.....	71
FIGURA 4-49 SECUENCIA FOCO 10.....	71
FIGURA 4-50 SECUENCIA FOCO 11.....	71
FIGURA 4-51 SECUENCIA FOCO 12.....	71
FIGURA 4-52 SECUENCIA FOCO 13.....	72
FIGURA 4-53 SECUENCIA FOCO 14.....	72
FIGURA 4-54 SECUENCIA FOCO 15.....	72
FIGURA 4-55 SECUENCIA FOCO 16.....	72
FIGURA 4-56 SECUENCIA FOCO 17.....	73
FIGURA 4-57 SECUENCIA FOCO 18.....	73
FIGURA 4-58 DETALLE APERTURA MÁXIMA DE PUERTA Y PINZA PARA LIBERACIÓN DE FOCO.....	73

FIGURA 5-1 SISTEMA DE MANIPULACIÓN DE PALOMITAS CON ESTRUCTURA PORTANTE VISTA DELANTERA .....	75
FIGURA 5-2 SISTEMA DE MANIPULACIÓN DE PALOMITAS CON ESTRUCTURA PORTANTE VISTA TRASERA.....	77
FIGURA 5-3 SISTEMA MÁQUINA DE PALOMITAS ESTADO PREVIO A ENTRADA A MÁQUINAS DE PALOMITAS EN CAMPO VISTA PERSPECTIVA .....	79
FIGURA 5-4 SISTEMA MÁQUINA DE PALOMITAS ESTADO PREVIO A ENTRADA A MÁQUINAS DE PALOMITAS EN CAMPO .....	79
FIGURA 5-5 SISTEMA DE PALOMITAS POSICIÓN Y ESTADO PREVIO A RECOLECCIÓN DE PALOMITAS EN CAMPO VISTA EN PERSPECTIVA .....	79
FIGURA 5-6 SISTEMA DE PALOMITAS POSICIÓN Y ESTADO PREVIO A RECOLECCIÓN DE PALOMITAS EN CAMPO .....	79
FIGURA 5-7 SISTEMA DE PALOMITAS CON RAMPAS RECOLECTANDO PALOMITAS EN CAMPO VISTA PERSPECTIVA. ....	80
FIGURA 5-8 SISTEMA DE PALOMITAS CON RAMPAS RECOLECTANDO PALOMITAS EN CAMPO VISTA FRONTAL .....	80
FIGURA 5-9 APERTURA PUERTAS PALOMITAS EN CAMPO VISTA LATERAL.....	80
FIGURA 5-10 APERTURA PUERTAS PALOMITAS EN CAMPO VISTA FRONTAL.....	80
FIGURA 5-11 AVANCE DE ROBOT PARA LIBERACIÓN DE VASO EN CAMPO .....	81
FIGURA 5-12 CIERRE DE PUERTAS DE PALOMITAS EN CAMPO.....	81
FIGURA 5-13 CIERRE DE PINZA IZQUIERDA EN CAMPO .....	81
FIGURA 5-14 CIERRE DE PINZA DERECHA EN CAMPO.....	81
FIGURA 5-15 RAMPAS EN POSICIÓN RECOGIDA VISTA SUPERIOR .....	82
FIGURA 5-16 RAMPAS EN POSICIÓN RECOGIDA VISTA PERSPECTIVA .....	82
FIGURA 5-17 RAMPAS EN POSICIÓN DE ENTRADA A RECOLECCIÓN VISTA SUPERIOR .....	82
FIGURA 5-18 RAMPAS EN POSICIÓN DE ENTRADA A RECOLECCIÓN VISTA PERSPECTIVA.....	82
FIGURA 5-19 RAMPAS EN POSICIÓN DE RECOLECCIÓN VISTA SUPERIOR .....	82
FIGURA 5-20 RAMPAS EN POSICIÓN DE RECOLECCIÓN VISTA PERSPECTIVA.....	82
FIGURA 5-21 SISTEMA DE PALOMITAS POSICIÓN Y ESTADO PREVIO A RECOLECCIÓN DE PALOMITAS EN CAMPO VISTA LATERAL .....	83
FIGURA 5-22 SISTEMA BANDEJA POSICIÓN RECOGIDA .....	84
FIGURA 5-23 SISTEMA BANDEJA POSICIÓN APERTURA 1.....	84
FIGURA 5-24 SISTEMA BANDEJA POSICIÓN APERTURA 2.....	84
FIGURA 5-25 SISTEMA BANDEJA POSICIÓN APERTURA DE RECOLECCIÓN .....	84
FIGURA 5-26 DETALLE PESTAÑA EN BANDEJA PARA BLOQUEO DE CAÍDA DE PALOMITAS DE RAMPA DE VASO DELANTERO .....	85
FIGURA 5-27 DETALLE CAÍDA PALOMITAS EN VASO TRASERO TRAS APERTURA DE BANDEJA.....	85
FIGURA 5-28 VISTA EN SECCIÓN DE SISTEMA DE APERTURA DE PUERTA DE PALOMITAS CON DETALLE DE CINTA DE TRACCIÓN PARA APERTURA CON GIRO DE PINZA DE VASO TRASERO.....	85
FIGURA 5-29 ANGULO DE PINZAS DE COMIENZO DE APERTURA DE PUERTAS.....	86
FIGURA 5-30 DETALLE APERTURA DE PUERTA DE PALOMITAS CON PUERTA SEMI-TRANSPARENTE .....	86
FIGURA 5-31 VISTA FRONTEL DE PUERTAS DE PALOMITAS ABIERTAS.....	87
FIGURA 5-32 VISTA LATERAL DE PUERTAS DE PALOMITAS ABIERTAS.....	87
FIGURA 5-33 MARGEN DE CAPTURA DE VASO CON PINZAS TRASERAS.....	87
FIGURA 5-34 DETALLE PINZA VASO TRASERO CERRADAS.....	88
FIGURA 5-35 DETALLE ORDEN DE CIERRE DE PINZAS DE VASO TRASERO.....	88
FIGURA 5-36 DETALLE APOYO INTERNO DE VASO TRASERO .....	89
FIGURA 5-37 SISTEMA VASO DELANTERO CON PARTES MÓVILES IDENTIFICADAS POR COLORES.....	89
FIGURA 5-38 SISTEMA VASO DELANTERO REPLEGADO VISTA LATERAL .....	90
FIGURA 5-39 SISTEMA VASO DELANTERO POSICIÓN PREVIA A CAPTURA VISTA LATERAL .....	91
FIGURA 5-40 SISTEMA VASO DELANTERO POSICIÓN PREVIA A CAPTURA VISTA PERSPECTIVA .....	91

FIGURA 5-41 SISTEMA VASO DELANTERO POSICIÓN DE CAPTURA VISTA LATERAL .....	91
FIGURA 5-42 SISTEMA VASO DELANTERO POSICIÓN DE CAPTURA VISTA PERSPECTIVA.....	91
FIGURA 5-43 SISTEMA VASO DELANTERO POSICIÓN DE VACIADO VISTA LATERAL .....	92
FIGURA 5-44 SISTEMA VASO DELANTERO POSICIÓN DE VACIADO VISTA PERSPECTIVA.....	92
FIGURA 6-1 <i>SISTEMA CIERRE DE CLAQUETAS Y ESTRUCTURA PORTANTE DE SUSTENTACIÓN</i> .....	93
FIGURA 6-2 BRAZO CIERRE CLAQUETA PLEGADO .....	94
FIGURA 6-3 BRAZO CIERRE CLAQUETA GIRO A 90° .....	94
FIGURA 6-4 BRAZO CIERRE CLAQUETA A 180°. ANGULO ABERTURA MÁXIMA .....	94
FIGURA 7-1 <i>VENTANA DE CÁLCULO DE POSICIÓN DE CENTRO DE MASAS DE SOLID EDGE</i> .....	100
FIGURA 7-2 <i>LOCALIZACIÓN DE CENTRO DE MASAS Y DISPOSICIÓN DE APOYOS DELANTEROS Y TRASEROS</i> .....	101
FIGURA 8-1 <i>TARJETA ELECTRÓNICA EUROBOTICS 2010</i> .....	115
FIGURA 8-2 <i>SENSOR E3Z-D61 DE OMRON</i> .....	116
FIGURA 8-3 <i>CIRCUITO DE SALIDA SENSOR E3Z-D61(OMRON, 2005)</i> .....	117
FIGURA 8-4 <i>SERVOMOTOR FUTABA S3003</i> .....	118
FIGURA 8-5 <i>SERVOMOTOR DYNAMIXEL AX12</i> .....	118
FIGURA 8-6 <i>CONEXIÓN A UART A TRAVÉS DE CIRCUITO CM-5 DE DYNAMIXEL</i> .....	119
FIGURA 8-7 <i>CONJUNTO MOTOR, REDUCTORA Y ENCODER MAXON</i> .....	120
FIGURA 8-8 <i>INTERFAX DE CONEXIONADO DEL ENCODER MAXON ENC20(MAXON)</i> .....	120
FIGURA 9-1 <i>DIAGRAMA DE CAPAS PRINCIPAL DE LA ARQUITECTURA SOFTWARE.</i> .....	124
FIGURA 9-2 <i>DIAGRAMA DE CAPAS SOFTWARE DEL CONTROL DE MECANISMOS BASADOS EN SERVOS ANALÓGICOS.</i> ....	127
FIGURA 9-3 <i>DIAGRAMA DE CAPAS SOFTWARE DEL CONTROL DE MECANISMOS BASADOS EN SERVOS AX12</i> .....	128
FIGURA 9-4 <i>DIAGRAMA DE CAPAS SOFTWARE DEL CONTROL DEL INTERCAMBIADOR DE STANDS</i> .....	129
FIGURA 9-5 <i>DIAGRAMA DE CAPAS SOFTWARE DE LA GESTIÓN DE MECANISMOS Y SISTEMAS MECÁNICOS</i> .....	131
FIGURA 9-6 <i>DIAGRAMA DE CAPAS SOFTWARE DE LA INTERFAZ DE CONTROL (MICROCONTROLADOR ESCLAVO)</i> .....	132
FIGURA 9-7 <i>DIAGRAMA DE CAPAS SOFTWARE DEL INTERFAZ DE CONTROL (MICROCONTROLADOR MAESTRO)</i> .....	132
FIGURA 1-1 <i>CONEXIÓN A UART A TRAVÉS DE CIRCUITO CM-5 DE DYNAMIXEL (USER'S MANUAL DYNAMIXEL AX-12, 2006)</i> .....	441
FIGURA 2-1 <i>ESQUEMA CONEXIONES DE MICRO AL RESTO DE ELEMENTOS DEL SISTEMA</i> .....	454
FIGURA 3-1 <i>ESQUEMA CONEXIONES A MÓDULOS PERIFÉRICOS DE MICROCONTROLADOR CON ETAPAS DE REMAPEADO DE PINES</i> .....	457
FIGURA 3-2 <i>ESQUEMA DE ANIDAMIENTO DE ESTRUCTURAS DE DATOS DEL PROYECTO</i> .....	461

# ÍNDICE DE TABLAS

TABLA 7-1 <i>CÁLCULO DETALLADO DE MASA TOTAL DEL DISEÑO</i> .....	100
TABLA 7-2 <i>LISTADO PIEZAS MECANIZADAS DE MATERIALES FENÓLICOS</i> .....	105
TABLA 7-3 <i>LISTADO PIEZAS MECANIZADAS DE METAL</i> .....	106
TABLA 7-4 <i>LISTADO PIEZAS TORNO</i> .....	107
TABLA 7-5 <i>LISTADO PIEZAS MECÁNICA</i> .....	109
TABLA 7-6 <i>LISTADO PIEZAS MECÁNICA</i> .....	110
TABLA 7-7 LISTADO TORNILLERÍA POR UNIÓN .....	111
TABLA 1-1 <i>COMPONENTES QUE DEBERÁN SER UTILIZADOS EN EL DISEÑO</i> .....	388
TABLA 1-1 <i>REMUNERACIÓN PERSONAL</i> .....	392
TABLA 2-1 <i>COSTE DE MATERIALES</i> .....	398
TABLA 2-2 <i>COSTE DE TORNILLERÍA</i> .....	399
TABLA 2-3 <i>COSTE DE PORTES</i> .....	400
TABLA 2-4 <i>COSTE DE MANO DE OBRA POR FASES DE PROYECTO</i> .....	400
TABLA 2-5 <i>COSTE DE AMORTIZACIÓN DE EQUIPOS</i> .....	401
TABLA 2-6 <i>COSTE DE AMORTIZACIÓN DE SOFTWARE</i> .....	401
TABLA 2-7 <i>TABLA TOTAL PRESUPUESTO EJECUCIÓN MATERIAL</i> .....	402
TABLA 4-1 <i>TABLA COSTE DE EJECUCIÓN POR CONTRATA</i> .....	402
TABLA 5-1 <i>IMPORTE TOTAL PROYECTO</i> .....	402
TABLA 1-1 <i>JUEGO DE INSTRUCCIONES DE CONTROL DE SERVOS DYNAMIXEL AX12 (ROBOTIS, 2006)</i> .....	443
TABLA 2-1 <i>CARACTERÍSTICAS DE CONEXIONES A PINES DE MICROCONTROLADOR UTILIZADOS</i> .....	455
TABLA 2-2 <i>PINES DE ENTRADA-SALIDA DE MÓDULOS PERIFÉRICOS DE MICROCONTROLADOR REMAPEADO</i> .....	456
TABLA 3-1 <i>TAREAS PROGRAMADAS EN SCHEDULER CON PERÍODO Y NIVEL DE PRIORIDAD DE EJECUCIÓN</i> ....	479
TABLA 3-2 <i>TIEMPOS DE RESPUESTA MÁXIMOS DEL DRIVER TC4427 MAS EL MOSFET IRF7389 (MICROCHIP TECHNOLOGY INC., 2006) (INTERNATIONAL RECTIFIER, 2004)</i> .....	494
TABLA 3-3 <i>BITS DE CÓDIGO DE ESTADO DE CONCLUSIÓN DE MOVIMIENTO DE SERVOS</i> .....	501
TABLA 3-4 <i>BITS DE CÓDIGO DE ESTADO DE CONCLUSIÓN DE MOVIMIENTO DE CARRO HORIZONTAL</i> .....	504



# ÍNDICE DE DIAGRAMAS

DIAGRAMA 1-1 <i>DIAGRAMA ORGANIZACIÓN TAREAS EQUIPO EUROBOT 2015.</i> .....	8
DIAGRAMA 8-1 <i>ARQUITECTURA HARDWARE. DIAGRAMA DE BLOQUES.</i> .....	113
DIAGRAMA 1-1 <i>DIAGRAMA BUCLE DE CONTROL DE MÓDULO "CONTROL_SYSTEM_MANAGER"</i> .....	426
DIAGRAMA 1-2 <i>DIAGRAMA FUNCIONAMIENTO FILTRO QUADRAMP.....</i>	434
DIAGRAMA 1-3 <i>GRÁFICA RESPUESTA A CONSIGNA ESCALÓN DE FILTRO QUADRAMP.....</i>	435
DIAGRAMA 1-4 <i>DIAGRAMA DE FLUJO DE FUNCIONES DE MÓDULO AX12.....</i>	444
DIAGRAMA 3-1 <i>DIAGRAMA DE INTERDEPENDENCIAS ENTRE ARCHIVOS DE CÓDIGO DEL PROYECTO</i> .....	463
DIAGRAMA 3-2 <i>DIAGRAMA DE FLUJO DE FUNCIÓN DE MÁQUINA DE ESTADOS PRINCIPAL.....</i>	466
DIAGRAMA 3-3 <i>DIAGRAMA DE FLUJO "TIPO" DE GESTIÓN DE ACTUADORES SIMPLES .....</i>	468
DIAGRAMA 3-4 <i>DIAGRAMA DE FLUJO "TIPO" DE FUNCIONES DE GESTIÓN DE ACTUADORES MÚLTIPLES.....</i>	470
DIAGRAMA 3-5 <i>DIAGRAMA DE FLUJO "TIPO" DE FUNCIONES DE EJECUCIÓN DE ACTUADORES MÚLTIPLES .....</i>	472
DIAGRAMA 3-6 <i>DIAGRAMA DE FLUJO "TIPO" DE FUNCIONES DE CONTROL DE ACTUADORES .....</i>	474
DIAGRAMA 3-7 <i>DIAGRAMA DE FLUJO DE TAREAS DEL PROYECTO .....</i>	477
DIAGRAMA 3-8 <i>DIAGRAMA DE FLUJO DE ORÍGENES DE NUEVO COMANDO .....</i>	485
DIAGRAMA 3-9 <i>DIAGRAMA DE ESQUEMATIZACIÓN DE LA CONFIGURACIÓN DEL MÓDULO DE CONTROL DE BUCLE PARA EL CONTROL DEL MOTOR DEL CARRO HORIZONTAL.....</i>	489



# ÍNDICE PLANOS

(00-01-01-001) BASE 1 1-2 .....	135
(00-01-01-001) BASE 1 2-2 .....	137
(00-01-02-014) TAPA SUPERIOR ESTRUCTURA.....	139
(00-01-03-007) ESTRUCTURA VERTICAL DELANTERO.....	141
(00-01-04-008) ESTRUCTURA VERTICAL TRASERO.....	143
(00-01-05-010) ESTRUCTURA LATERAL IZQUIERDO.....	145
(00-01-06-011) ESTRUCTURA LATERAL DERECHO .....	147
(00-01-07-025) ESTRUCTURA TAPA BLOQUE MOTOR.....	149
(00-01-08-036) ESTRUCTURA SOPORTE SERVO BRAZOS PALOMITAS .....	151
(00-01-09-038) ESTRUCTURA SOPORTE SERVO BRAZOS PALOMITAS DERECHO.....	153
(00-01-10-084) TAQUITOS DE ALUMINIO .....	155
(00-01-11-003) REFUERZO DELANTERO BASE .....	157
(00-01-12-012) ESTRUCTURA CHAFLAN TRASERO_MIR .....	159
(00-01-13-027) ESTRUCTURA PISO DELANTERO .....	161
(00-01-14-026) ESTRUCTURA PISO TRASERO 1.....	163
(00-01-15-028) ESTRUCTURA PARED VASO .....	165
(00-01-16-029) ESTRUCTURA PENDIENTE PISO TRASERO 2.....	167
(00-01-17-089) PALANCA APERTURA TRAMPILLAS INFERIORES.....	169
(00-01-18-203) SOPORTE TEFLON APOYO TRASERO .....	171
(00-01-19-054) SEPARADOR TRASERO BASE CARROS Y BASE PUERTA .....	173
(00-01-20-055) SEPARADOR LASTRE Y BASE CARROS DELANTERO .....	175
(00-01-21-053) SEPARADOR BASE Y BASE CARROS DELANTERO .....	177
(00-01-22-050) SOPORTE ROLLON.....	179
(00-02-01-013) TRAMPILLA TRASERA INFERIOR .....	181
(00-02-02-017) TRAMPILLA TRASERA SUPERIOR .....	183
(01-00-00-E01) BLOQUE MOTOR EXPLOSION .....	185
(01-00-01-000) PLACA ENCODER IZQUIERDO .....	187
(01-00-02-000) PLACA MOTOR IZQUIERDA .....	189
(01-00-03-000) LLANTA .....	191
(01-00-04-000) RUEDA FOAM ANCHO 40 .....	193
(02-00-01-006) TRASERO .....	195
(02-00-02-006) TRASERO SUPERIOR .....	197

(02-00-03-005) DELANTERO .....	199
(02-00-04-204) SUPLEMENTO TRASERO .....	201
(03-00-00-E01) BALANCIN ENCODER EXPLOSION .....	203
(03-00-01-030) BISAGRA EXTERIOR BALANCIN ENCODER .....	205
(03-00-02-031) BISAGRA INTERIOR BALANCIN ENCODER .....	207
(03-00-03-032) BRAZO EXTERIOR BALANCIN ENCODER .....	209
(03-00-04-033) BRAZO INTERIOR BALANCIN ENCODER .....	211
(03-00-05-000) RUEDA BALANCIN ENCODER.....	213
(03-00-06-065) SEPARADOR BRAZOS ENCODER.....	215
(03-00-07-066) EJE BRAZOS ENCODER .....	217
(04-00-00-C01) CONJUNTO COMPLETO SISTEMA STANDS .....	219
(04-00-01-009) BASE CARROS IZQUIERDA.....	221
(04-00-02-093) TAPA BISAGRA SUPERIOR.....	223
(04-00-03-094) GUIAS 6 STAND .....	225
(04-00-04-095) GUIAS 8 CARROS.....	227
(04-01-00-E01) CARRO PINZA 2 DERECHA EXPLOSION .....	229
(04-01-01-106) BASE 1 CARRO PINZA .....	231
(04-01-02-107) BASE 3 CARRO PINZA .....	233
(04-01-03-108) BASE 2 CARRO PINZA .....	235
(04-01-04-109) BASE 4 CARRO PINZA .....	237
(04-01-05-110) PINZA SUPLEMENTO.....	239
(04-01-06-111) PINZA PALANCA .....	241
(04-01-07-112) PINZA LEVA .....	243
(04-01-08-113) CABEZA SERVO CARRO PINZA .....	245
(04-01-09-114) ALOJAMIENTO EJE SERVO .....	247
(04-01-10-115) SEPARADOR LATERAL BASE 1 CARRO PINZA .....	249
(04-01-11-116) SEPARADOR CENTRAL BASE 1 CARRO PINZA .....	251
(04-01-12-117) SEPARADOR LATERAL BASE 2 CARRO PINZA .....	253
(04-01-13-118) SEPARADOR CENTRAL BASE 2 CARRO PINZA .....	255
(04-01-14-119) SEPARADOR LATERAL BASE 3 CARRO PINZA .....	257
(04-01-15-120) SEPARADOR CENTRAL BASE 3 CARRO PINZA .....	259
(04-02-00-E01) CARRO SERVO 2 DERECHA EXPLOSION.....	261
(04-02-01-096) PALANCA CARRO SERVO .....	263

(04-02-02-097) PLATO CARRO SERVO LADO SOPORTE EJE .....	265
(04-02-03-098) PLATO CARRO SERVO LADO SERVO .....	267
(04-02-04-099) SEPARADOR PLATOS CARRO SERVO .....	269
(04-02-05-100) TAPA CARRO SERVO DERECHA _MIR1 .....	271
(04-02-06-101) BASE CARRO SERVO DERECHA _MIR1 .....	273
(04-02-07-102) SOPORTE SERVO CARRO SERVO .....	275
(04-02-08-103) SOPORTE EJE SERVO CARRO SERVO .....	277
(04-02-09-104) SEPARADOR LATERAL CARRO SERVO .....	279
(04-02-10-105) SEPARADOR CENTRAL CARRO SERVO .....	281
(05-00-00-E01) CONJUNTO PALETAS EXPLOSIÓN .....	283
(05-00-01-002) INFERIOR ESTRUCTURA PALETAS .....	285
(05-00-02-034) MEDIO ESTRUCTURA PALETAS .....	287
(05-00-03-035) SUPERIOR ESTRUCTURA PALETAS .....	289
(05-00-04-056) SEPARADOR ESTRUCTURA TI .....	291
(05-00-05-057) SEPARADOR ESTRUCTURA TS .....	293
(05-00-06-058) SEPARADOR ESTRUCTURA DI .....	295
(05-00-07-059) SEPARADOR ESTRUCTURA DS .....	297
(05-00-08-062) BRIDA PALETAS .....	299
(05-00-09-063) PALETA .....	301
(06-00-00-E01) CONJUNTO CARRO HORIZONTAL .....	303
(06-00-01-004) GUIA CARRO HORIZONTAL .....	305
(06-00-02-051) BRAZO CARRO HORIZONTAL .....	307
(06-00-03-067) SOPORTE MOTOR CARRO HORIZONTAL .....	309
(06-00-04-070) TENSOR INFERIOR CORREA .....	311
(07-00-01-020) BASE PUERTA INFERIOR .....	313
(07-00-02-047) BRAZO PUERTA SUPERIOR .....	315
(07-00-03-048) PUERTA .....	317
(07-00-04-079) LEVA SERVO PUERTA .....	319
(07-00-05-200) SOPORTE SERVO INFERIOR .....	321
(08-00-00-E01) CONJUNTO PINZA VASO DELANTERO .....	323
(08-00-01-021) RAMPA BASE .....	325
(08-00-02-022) SOPORTE DYNAMIXEL FIJO LATERAL .....	327
(08-00-03-023) SOPORTE DYNAMIXEL FIJO LATERAL DERECHO .....	329

(08-00-04-024) SOPORTE DYNAMIXEL FIJO .....	331
(08-01-01-041) APOYO INFERIOR.....	333
(08-01-02-042) APOYO SUPERIOR.....	335
(08-01-03-043) BRAZO SOPORTE DYNAMIXEL MOVIL .....	337
(08-01-04-044) BRAZO ESTABILIZADOR SOPORTE DYNAMIXEL MÓVIL .....	339
(08-01-05-045) SEPARADOR DYNAMIXEL MÓVIL.....	341
(08-01-06-046) PINZA .....	343
(09-00-01-071) DISCO BRAZO CLAQUETA.....	345
(09-00-02-000) VARILLA BRAZO CLAQUETA .....	347
(10-01-00-E01) CONJUNTO BRAZOS PALOMITAS EXPLOSIÓN .....	349
(10-01-01-037) ESTRUCTURA SOPORTE SERVO DYNAMIXEL BRAZOS PALOMITAS.....	351
(10-01-02-039) ESTRUCTURA SOPORTE SERVO DYNAMIXEL BRAZOS PALOMITAS DERECHO ....	353
(10-01-03-083) BASE BRAZOS PALOMITAS 2 .....	355
(10-01-04-082) PENDIENTE BRAZOS PALOMITAS 2 .....	357
(10-01-05-081) TAPA BRAZOS PALOMITAS 2 .....	359
(10-01-06-085) SUPLEMENTO SERVO BRAZOS PALOMITAS .....	361
(11-00-01-086) PINZA COMPLETA.....	363
(11-00-02-087) PINZA PARTIDA EJE .....	365
(11-00-03-088) PINZA PARTIDA PUNTA .....	367
(12-00-01-075) LEVA SERVO TRAMPILLA SUPERIOR 2 .....	369
(12-00-02-076) BRAZO SERVO TRAMPILLA SUPERIOR 2 .....	371
(12-00-03-077) BISAGRA TRAPILLA SUPERIOR .....	373
(12-00-04-201) SOPORTE SERVO TRAMPILLA SUPERIOR .....	375
(14-00-01-015) TAPA ELECTRÓNICA.....	377
(14-00-02-018) TAPA TRASERA ELECTRÓNICA.....	379
(14-00-03-040) SOPORTE ELECTRÓNICA .....	381
(14-00-04-078) SEPARADOR SOPORTE ELECTRÓNICA.....	383

## **RESUMEN**

El presente proyecto tiene como objetivo crear un robot mecánicamente capaz de homologarse y participar en Eurobot 2015, solventando todos los retos que se plantean para esta prueba.

El trabajo se ha desarrollado en colaboración con la asociación de robótica de Coslada y la IEEE de la UAH, en un área perfectamente acotada.

El proyecto engloba el diseño mecánico y la fabricación del robot primario y la adaptación de la electrónica dedicada para el control de los sistemas de manipulación de elementos de juego, incluyendo la programación para tal fin y para la comunicación con la electrónica de alto nivel.



## **SUMMARY**

This project aims to create a robot that can be homologated and can participate in Eurobot 2015, resolving all the challenges for this test.

The work was developed in collaboration with the association of robotics IEEE Coslada and UAH , in an area bounded perfectly.

The project includes mechanical design and manufacture the primary robot and the adaptation of dedicated electronic control systems handling of game elements, including programming for that purpose and to communicate with high-level electronics.



## **PALABRAS CLAVE**

Robot

Eurobot

Diseño mecánico



## RESUMEN EXTENDIDO

Este proyecto tiene su origen en Eurobot y en concreto en la edición 2015 con la prueba denominada Robomovies.

Eurobot es una competición de robots autónomos, promovida por diferentes organizaciones, entidades y comités nacionales cuyo objetivo fundamental radica en fomentar la robótica educativa para los jóvenes estudiantes. Se ha creado un entorno atractivo que con el pretexto de una competición desemboca finalmente en una convivencia amistosa con un intercambio constante de experiencias, todo tipo de consejos y conocimientos de ingeniería en torno a un reto común. Siendo el resultado final un enriquecimiento técnico y cultural.

La normativa de esta prueba año tras años fomata la creatividad de los participantes, planteando un nuevo reto, que aunque manteniendo una base común formada por las características el campo, el tamaño de los robots y otros elementos como la estructura del sistema de balizamiento, introduce una serie de nuevos elementos de juego y objetivos a conseguir que además de hacer de la creatividad una necesidad, iguale las posibilidades de los nuevos participantes frente a los veteranos.

En la edición 2015 se ha presentado una nueva normativa tematizada en el mundo del cine, en la que se han introducido diferentes elementos característicos de él. Se enumeran brevemente a continuación las características fundamentales para comprender el resto del contexto de este resumen. En la imagen Figura 1-1 se muestra una recreación del campo de la prueba.



Figura 1-1 Campo Eurobot 2015, con elementos de juego

- Los stands son cilindros de madera que deberán ser recogidos de posiciones conocidas, apilando 8 con una pelota de tenis en su parte superior.
- Los vasos de palomitas tienen palomitas en su interior (esferas de poliestireno expandido de 40 mm de diámetro) y deberán ser recogidos de posiciones conocidas y transportados hasta otras posiciones.
- Las máquinas de palomitas son dispensadores de palomitas que deberán ser recogidas y depositadas en el interior de vasos u otras zonas del campo.
- Las claquetas son las típicas pizarras utilizadas en el mundo del cine para separar las diferentes "tomas", el objetivo es cerrar las de tu color.

Aunque esta prueba permite para su participación una resolución parcial de los problemas planteados y un nivel de complejidad adaptable a los esperables para la elaboración de un proyecto de este tipo, se ha acometido dicha empresa como integrante del equipo de competición “Eurobotics Engineering” formado por miembros de la asociación de robótica de Coslada y la IEEE de la UAH. Este equipo que compite al más alto nivel de esta prueba con una experiencia de más de 12 años, tiene un sistema de organización del trabajo con una división de tareas, que ha permitido acotar el trabajo realizado en este proyecto del resto del equipo. En el Diagrama 1-1 se ha esquematizado la división del trabajo creada para los diferentes integrantes.

Este proyecto según se ha planteado no ha pretendido alcanzar un alto nivel de complejidad en ninguna materia concreta, su objetivo se ha centrado en abarcar el máximo número de materias y enfocarlo en la interdisciplinariedad y la creatividad, siendo éstas las características fundamentales de esta prueba. Para conseguirlo se han asumido las tareas de diseño íntegro de la mecánica y el proceso productivo del robot "Petinto", la adaptación de la electrónica de control de los sistemas de manipulación de elementos de juego y la programación de dicha electrónica.

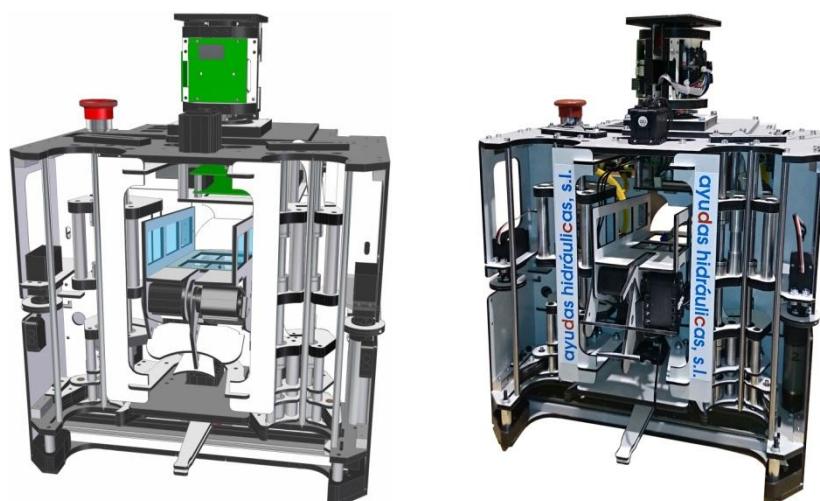


Figura 1-2 Robot "Petinto" vista delantera Diseño 3D a la izquierda, Fotografía a la derecha

Previa a la realización de las fases descritas en esta memoria ha existido una etapa inicial colaborativa con el resto del equipo, en la que de forma conjunta y con las aportaciones de la experiencia común se han definido las líneas a seguir y los objetivos funcionales mínimos, hasta concretar las especificaciones del diseño mecánico y el protocolo de comunicación con la estructura de datos que compartirán los dos microcontroladores que controlaran todo el sistema. Todo el control del robot se va a realizar desde una misma tarjeta diseñada por “Eurobotics Engineering” para anteriores participaciones en esta prueba y que integra dos microcontroladores para la división de las tareas de control.

La memoria del proyecto está organizada en el mismo orden en el que han sido realizadas las diferentes fases de diseño. Estas son: el diseño mecánico con la generación de toda la documentación necesaria para la fabricación, la adaptación de la electrónica compartida con el resto de sistemas y la programación para el control de los sistemas

mecánicos diseñados. A continuación se describe el trabajo desarrollado en cada una de estas etapas.

## EL DISEÑO MECÁNICO

El diseño mecánico que se ha realizado ha tenido como base las especificaciones que fueron concretadas junto con el resto del equipo en la fase inicial del proyecto. Era muy importante definir con suficiente precisión el proyecto para poder trabajar desde un primer momento a todos los niveles de forma paralela.

Las especificaciones del diseño mecánico surgieron de la definición de una estrategia de juego, que tenía como base una secuencia de movimientos y operaciones que se consideró óptima para la consecución de la máxima puntuación, sin interferencias de los robots oponentes. La mayoría de las especificaciones mecánicas fueron los requerimientos mínimos para la ejecución de esta secuencia. Estas especificaciones han abarcado desde la definición de los sistemas necesarios y su ubicación, hasta la velocidad de procesado de los elementos de juego. Otro punto importante de estas especificaciones incluía la reutilización de la mayor parte de los sistemas de accionamiento, servos y motores, de que disponía el equipo de anteriores participaciones en esta prueba.

La primera fase del diseño mecánico ha sido probablemente la más importante y también quizás la menos documentada debido a la complejidad por su carácter en parte conceptual. En esta fase se ha estudiado cada uno de los sistemas a implementar de manera conjunta, aproximando cada uno de ellos a su diseño definitivo, mediante la creación de prototipos rápido y bocetos en 2D, pero siempre ubicándolos dentro del conjunto del robot en torno a los demás sistemas. El objetivo final de esta fase ha sido ubicar y acotar de una manera aproximada el volumen disponible de cada uno de los sistemas, definiendo su funcionamiento y estructura aproximada. En definitiva lo que se ha intentado evitar es desarrollar sistemas de forma completamente definida que luego condicione el desarrollo de otros.

Concluida esta fase y con una aproximación bastante cercana al diseño definitivo de cada uno de los sistemas, se ha diseñado la estructura que conforma el esqueleto del robot y que va a dar soporte a cada uno de los sistemas a implementar. Previo a este diseño se han planteado una serie de consideraciones, basadas en la experiencia adquirida en anteriores participaciones en esta prueba, sobre las características constructivas que debía reunir la estructura a implementar y las tecnologías de fabricación que iban a ser utilizadas. Por supuesto los sistemas de fabricación empleados han sido determinantes para la tipología de las estructuras creadas. Como punto destacable en el diseño de la estructura se ha prestado especial atención a la distribución de masas, intentando que el centro de masas resultante se localizara lo más bajo posible y próximo al centro geométrico del robot en una vista en planta, siendo el objetivo permitir la máxima aceleración en los movimientos de desplazamiento.

El diseño de la estructura se ha realizado de manera conjunta con el sistema de tracción empleado y el sistema de control de posición por odometría. El sistema de tracción utilizado ha sido un sistema bidireccional, y se ha partido para el diseño de dos motores brushless reutilizados para reducir el coste final del proyecto. Ha sido necesario dimensionar una etapa

de reducción teniendo en cuenta la potencia transmitida para adaptar las revoluciones del motor al rango total de velocidades aplicables para el desplazamiento en la prueba. En este punto se han documentado todos los cálculos y graficas utilizadas para la elección de los elementos del sistema de transmisión elegido y los realizados para el cálculo exacto de la distancia entre ejes. Finalizada la etapa de diseño se ha comprobado que el motor utilizado, al ser una imposición, tiene la potencia necesaria para generar las aceleraciones definidas en las especificaciones de diseño. Para realizar este estudio tratándose de un motor brushless con controladora incluida, se ha utilizado la curva del momento en función de las revoluciones del motor a tensión nominal, para calcular la aceleración tangencial máxima de la rueda (aceleración del robot) en función de la velocidad de avance de este, y a partir de esta expresión la velocidad en función del tiempo para la máxima aceleración.

El sistema de control odométrico diseñado básicamente ha consistido en un brazo pivotante amortiguado, en el que se ha prestado especial importancia a la precisión del sistema constructivo para minimizar los errores sistemáticos de captación entre los dos encoder. También han sido analizadas las características constructivas implementadas para reducir los no sistemáticos.

Concluida la parte estructural se ha continuado el diseño de cada uno de los sistemas de manipulación que se han integrado en el robot, siguiendo unas etapas de diseño comunes para todos ellos, pero según los requerimientos del sistema, incluyendo si se han considerado necesarios diferentes cálculos, bien para asegurar que se cumplían las especificaciones de tiempo de procesado requeridas o para comprobar que los sistemas de accionamientos estaban correctamente dimensionados.

Dentro de los sistemas de manipulación diseñados el más destacable puede ser el subsistema de elevación del sistema de manipulación de stand. Este sistema ha necesitado de una fase previa de estudio más amplia que el resto, pasando por la construcción de diferentes prototipos operativos y finalmente una amplia fase de cálculo. Los cálculos realizados para este caso han sido comprobados mediante un programa de simulación de mecanismo.

Una vez finalizados todos los diseños se ha generado toda la documentación necesaria para el proceso productivo, incluyendo desde lista de piezas según proceso de fabricación hasta el listado de tornillería detallada y componentes eléctricos. Para la codificación de los planos generados se ha definido un sistema de numeración para una fácil identificación por conjuntos, subconjuntos, piezas y código de montaje.

## HARDWARE

La electrónica de control de los sistemas mecánicos está basada en una arquitectura compuesta por dos microcontroladores de 16 bits de la familia dsPIC de Microchip. El microcontrolador maestro está destinado al control de tracción, evitación de obstáculos y estrategia de juego del robot, y el microcontrolador esclavo destinado al control de los sistemas mecánicos.

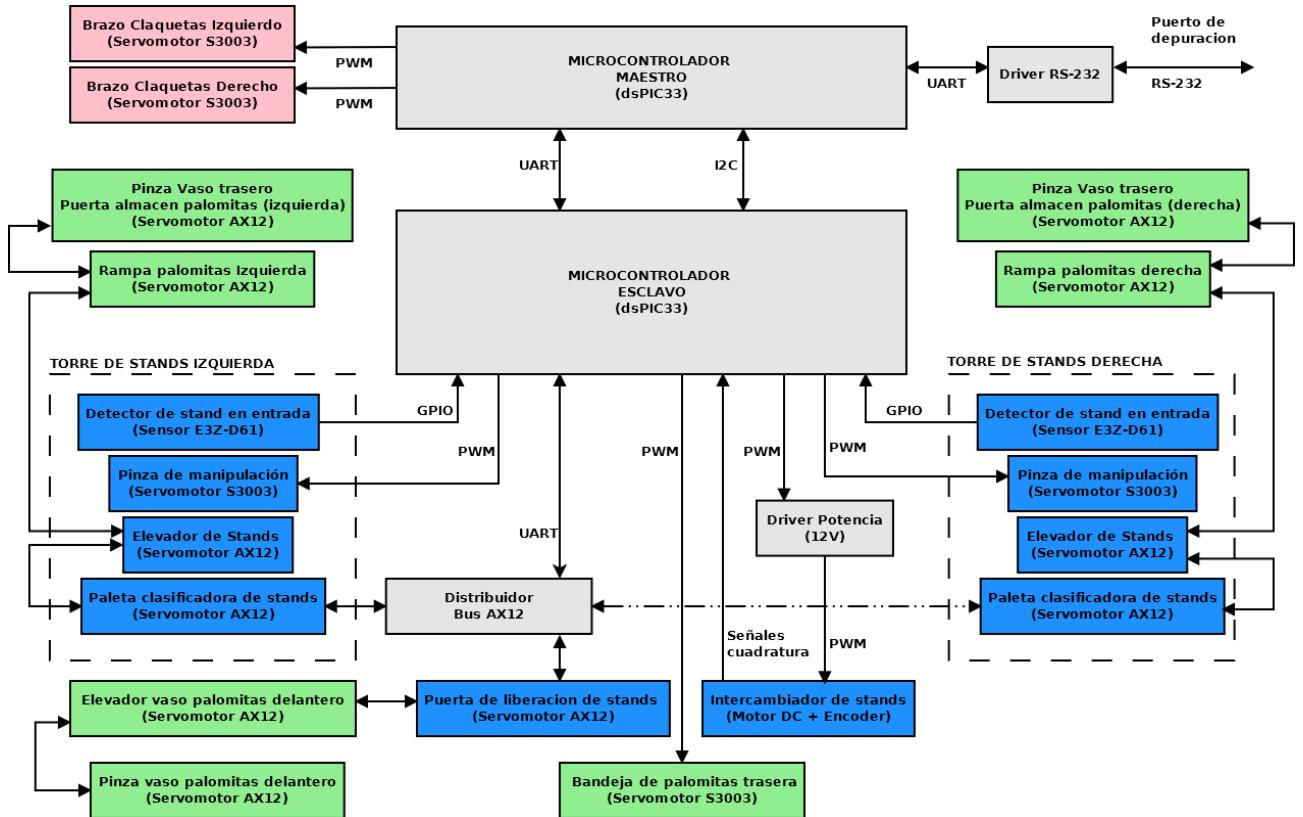


Figura 1-3 Arquitectura hardware. Diagrama de bloques.

Como se observa en la Figura 1-3, ambos microcontroladores se encuentran comunicados a través de un bus I2C maestro-esclavo y un bus serie tipo UART full-duplex.

Cada microcontrolador dispone de una interfaz de depuración serie full-duplex, accesible mediante un bus de niveles RS-232. Dicha interfaz se encuentra conectada a uno de los módulos UART del microcontrolador maestro mediante un transceiver RS-232/UART. Realizando un bypass software hacia el segundo módulo UART del dsPIC maestro, es posible acceder a la interfaz de depuración del microcontrolador esclavo.

Por otro lado, la interfaz I2C implementa la interfaz principal de control y monitorización de los sistemas conectados al microcontrolador esclavo.

Según se muestra en la Figura 1-3, el microcontrolador maestro controla directamente los actuadores del sistema de claquetas, mientras que el resto de sistemas son gestionados desde el microcontrolador esclavo.

Para la detección de stands dentro del rango de las paletas clasificadoras se han utilizado sensores digitales de luz difusa E3Z-D61 de Omron. Dichos sensores son leídos por el microcontrolador mediante pines tipo GPIO.

Las pinzas de manipulación de stands, la bandeja de palomitas trasera y los brazos utilizados para cerrar claquetas utilizan servomotores o "servos" Futaba S3003. Se trata de servos analógicos que son controlados mediante una señal PWM de 20Hz generada por el microcontrolador a partir de un periférico HW dedicado.

El resto de mecanismos, excepto el intercambiador de stands entre torres, utilizan servos digitales Dynamixel AX12. Estos servos se controlan mediante un bus serie half-duplex que es compartido por todos los servos que estén conectados al bus. Así, para el control de los servos AX12 se utiliza uno de los módulos UART del dsPIC esclavo. Mediante un distribuidor cableado, se generan 3 ramas que interconectan todos los servos AX12.

Por último, el intercambiador de stands utiliza un motor de continua alimentado a 12V mediante un driver de potencia, y controlado mediante una señal PWM de 19Khz. Dicha señal es generada desde el microcontrolador utilizando un módulo HW generador de señales PWM. El motor cuenta con un encoder en cuadratura fijado en su eje trasero, y cuyas señales son decodificadas mediante un módulo HW específico del microcontrolador obteniendo hasta 4 veces el número de pulsos por vuelta de encoder.

## DISEÑO SOFTWARE

Los sistemas mecánicos se encuentran gestionados por un software que permite sincronizar actuadores y sensores con el fin de generar secuencias de movimientos complejas y precisas.

El software desarrollado utiliza o se apoya en las librerías Open Source Aversive4dsPIC desarrolladas por el equipo Eurobotics Engineering, a partir de las librerías Aversive. Estas últimas, fueron inicialmente desarrolladas por el equipo de Eurobot Microb Technology para microcontroladores AVR de Atmel. Posteriormente el equipo Eurobotics Engineering añadió la compatibilidad con microcontroladores dsPIC de Microchip.

Se identifican 3 sistemas que han de funcionar de forma concurrente:

- Sistema de palomitas
- Sistema de stands
- Sistema de claquetas

Cada sistema está compuesto por sensores y mecanismos que combinados y correctamente sincronizados, permiten ejecutar secuencias que implementan diferentes modos de funcionamiento de cada sistema.

El software desarrollado se divide en capas de abstracción, cada capa puede contener varios bloques que utilizan o se apoyan en bloques de la capa inferior.

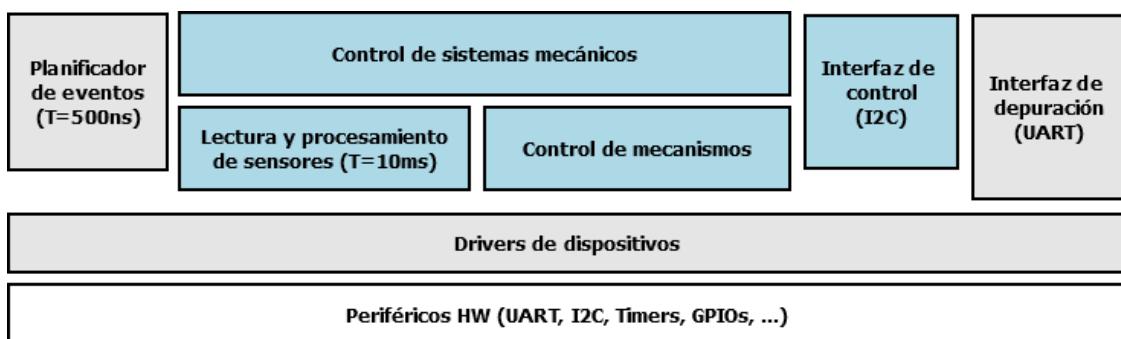


Figura 1-4 Diagrama de capas principales de la arquitectura software

El diagrama de capas principal del software del microcontrolador esclavo se muestra en la Figura 1-4. Los bloques grises se identifican con módulos utilizados de las librerías Aversive4dsPIC, en azul se representan los bloques desarrollados en este proyecto y en blanco los módulos HW del microcontrolador.

El bloque de *control de sistemas mecánicos* gestiona los diferentes modos de funcionamiento de los sistemas de stands y palomitas. Para ello, se apoya sobre el bloque de *lectura y procesamiento de sensores* y el de *control de mecanismos*. Estos tres bloques son accesibles y controlables a través de una *interfaz de control I2C*, utilizada por el microcontrolador maestro para sincronizar estos sistemas mecánicos con movimientos del robot. En las secciones siguientes se describen en detalle cada uno de estos bloques.



# **1. Introducción y Objetivos**

---

## **1.1. Introducción**

---

En este primer capítulo se describe el entorno en el que se encuentra enmarcado el proyecto desarrollado. Se explica el origen de la competición de Eurobot y su filosofía, y a continuación se hace una introducción a la prueba que ha originado este proyecto. De forma paralela se describe la motivación y los antecedentes para la elección de este proyecto.

La competición Eurobot da la oportunidad de experimentar en múltiples disciplinas y áreas de la tecnología aplicadas a la competición con robots autónomos.

En la última parte de este capítulo se exponen los objetivos que se desean conseguir con la realización de este proyecto, enmarcado dentro de un trabajo en equipo, así como la estructura de la presente memoria.

## **1.2. Eurobot**

---

Creado en 1998, a partir de la Copa de Francia de robots nacida en 1992, Eurobot es un concurso internacional de aficionados a la robótica que permite el encuentro de equipos de jóvenes, clubes independientes o grupos de estudiantes, con ganas de llevar a cabo un proyecto de ingeniería aplicada.

Hasta el año 2004 las finales de Eurobot se venían celebrando en La Ferté-Bernard, un pueblo de Francia, a partir de dicha fecha cada año se celebra en un país diferente de Europa. Aunque Eurobot tiene lugar en Europa, países de otros continentes también han participado en el concurso. En 2004, 21 países, representados por 205 equipos se implicaron en este concurso científico y técnico pasando por unas calificaciones nacionales y una final internacional.

En el año 2015 ha sido necesaria la celebración de 11 pruebas de calificación nacional, entre las que se ha encontrado la española para la selección de los tres equipos a los que la organización de Eurobot limita el acceso al certamen final.

### **1.2.1. El concurso**

---

Eurobot es un concurso amateur internacional de robótica abierto a equipos de gente joven, organizados, bien como proyectos de estudiantes o bien en clubes independientes. Un equipo se compone de varias personas unidas en un proyecto común.

Los valores de Eurobot se basan en el juego limpio, la solidaridad, el compartir conocimientos técnicos y la creatividad, a través de técnicas y gestión de proyectos más que de competición. Los objetivos del concurso son interesar al mayor público posible en robótica

y fomentar la práctica en grupo de la ciencia entre los jóvenes. Eurobot y sus clasificaciones nacionales tienen la intención de llevarse a cabo dentro de un espíritu amistoso y deportivo. Por lo tanto, más que un concurso de ingeniería para gente joven, Eurobot es un pretexto amistoso para la imaginación técnica y el intercambio de ideas, consejos y conocimientos de ingeniería en torno a un reto común. La creatividad se desarrolla y la interdisciplinariedad es requerida. El enriquecimiento técnico y cultural es el objetivo. En esta competición todo el mundo gana en conocimiento, experiencia y amistad.

Eurobot se lleva a cabo en Europa, pero está abierto a otros países. Los países que presenten a más de tres equipos deben organizar una competición local para la selección de tres equipos, límite máximo que impone la organización del certamen final. Eurobot y sus selecciones nacionales son atendidos mayoritariamente por voluntarios de todas las nacionalidades que creen en el valor educacional de esta experiencia, y son casi todos antiguos participantes.

Cada año, las reglas se publican alrededor de octubre. Los equipos tienen entonces varios meses para diseñar y construir sus robots y competir entre abril y junio en el concurso nacional, en el cual se pueden seleccionar sólo 3 equipos por país. Estos mejores equipos de cada país se reunirán después para competir en las finales.

Las reglas cambian cada año para dar a los nuevos participantes el mismo estatus que a todos y estimular a los que ya han participado con anterioridad. Sin embargo, la base es siempre la misma:

- Los robots tienen aproximadamente el tamaño de un cubo de 30 cm.
- El campo de juego tiene una dimensión de 2 metros de ancho y 3 metros de largo.
- El juego tiene 90 segundos de duración.
- Los robots tienen que llevar a cabo varias tareas en el campo con el fin de ganar tantos puntos como sea posible.
- Los robots tienen que evitarse unos a otros. Los equipos pueden colocar balizas en el otro robot para averiguar su ubicación.
- Los robots son completamente autónomos y deben hacer todas las acciones por ellos mismos con la ayuda de sensores.

### 1.2.2. La asociación de Eurobot

La asociación de Eurobot es una organización internacional creada en mayo de 2004 y registrada en Francia. Nació 6 años después de la competición de Eurobot para estructurar la organización de este concurso en su crecimiento, y para favorecer el espíritu de intercambio y cooperación entre los diferentes organizadores.

Los miembros de la asociación están compuestos por los siguientes grupos:

- Los 3 miembros fundadores: Planète Sciences, el Grupo VM y la ciudad La Ferté Bernard (Francia).
- Los comités organizadores nacionales de las clasificaciones de Eurobot:
  - Francia - desde 1993
  - Suiza - desde 1998
  - Bélgica - desde 2001
  - Serbia & Montenegro - desde 2002
  - España - desde 2002
  - Austria - desde 2004
  - República Checa - desde 2005
  - Argelia - desde 2005
  - Italia - desde 2005
  - Alemania - desde 2005
  - Rumanía - desde 2006
  - Reino Unido - desde 2006
- Miembros beneficiarios, por ejemplo clubes en países sin comités organizadores nacionales.
- Miembros individuales.

Además existe un Comité Ejecutivo que se encarga de dirigir y controlar las operaciones del día a día de Eurobot.

Para llevar la competición de Eurobot, y a parte de las organizaciones nacionales, se forman grupos internacionales. Entre ellos el Comité de Árbitros gestiona las reglas, y el Grupo Operativo de Eurobot gestiona los equipos sin clasificaciones nacionales.

### 1.3. Origen del proyecto

Este proyecto tiene su origen en la edición Eurobot 2015 con la prueba Robomovies. Se ha colaborado con el equipo de competición “Eurobotics Engineering” formado por miembros de la asociación de robótica de Coslada y la IEEE de la UAH para el desarrollo de dos robots "Petinto" ver Figura 1-1 y Figura 1-2, y "Tirantes" con el fin de competir en dicha prueba.

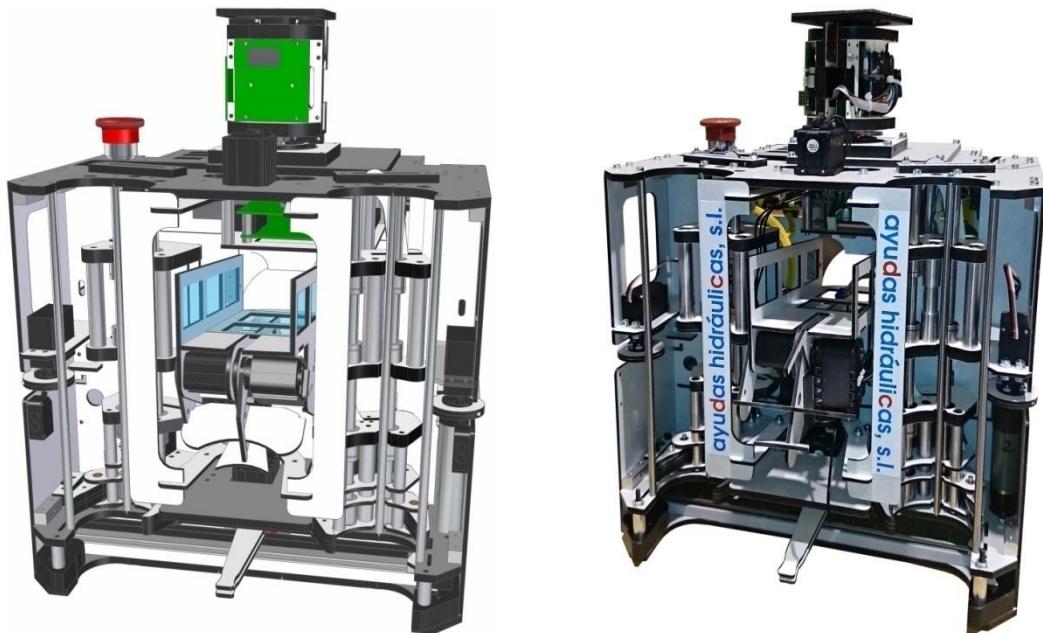


Figura 1-1 Robot "Petinto" vista delantera Diseño 3D a la izquierda, Fotografía a la derecha

La asociación de robótica de Coslada que es la promotora originaria del equipo “Eurobotics Engineering” está formada por antiguos alumnos de la universidad de Alcalá de Henares con una amplia experiencia en la participación de la prueba de Eurobot y en otros certámenes de robótica educativa.

Personalmente he participado junto con algunos de ellos, en los diferentes proyectos relacionados mayoritariamente con Eurobot y que se han ido sucediendo desde el año 2003. Inicialmente todo esta inquietud surgió a raíz del proyecto promovido por el profesor D. Julio Pastor Mendoza y bajo la financiación del Departamento de Electrónica de la Universidad de Alcalá de Henares. Durante estos años siempre he tenido alguna relación pero con muy diferentes grados de implicación, enmarcando mí trabajo casi siempre dentro del área mecánica.

A continuación se enumeran las diferentes participaciones en la prueba de Eurobot.

- Eurobot 2003 "Cara o cruz" La Ferté-Bernard (Francia). Robot Husillo (UAH). Fabricación mecánica.
- Eurobot 2004 "Rugbi del coco" La Ferté-Bernard (Francia). Robots Electrococo y Joselito(UAH). Diseño y fabricación mecánica.

- Eurobot 2005 "Juego de bolos" Yverdon les Bains (Suiza). Robot Campanolo (UAH). Todos los campos. Nominado para el premio de “Mejor concepto de robot”.
- Eurobot 2006 "Un divertido golf". Robot Caponata (UAH). Apoyo técnico fabricación mecánica.
- Eurobot 2007 "Recycling" La Ferté-Bernard (Francia). Robot Mr. Proper (UAH). Apoyo técnico fabricación mecánica.
- Eurobot 2008 "Mission to Mars" Heidelberg (Alemania). Robot Topolino (UAH). Apoyo técnico fabricación mecánica. Premio de “Mejor concepto de robot”.
- Eurobot 2010 "Feed the World" Rapperswil-Jona (Suiza). Robot Primos (Proyecto independiente). Todos los campos.
- Eurobot 2011 "Chess`up!" Astrakhan (Rusia). Robot Zamorano (Eurobotics Engineering-UAH). Apoyo técnico fabricación mecánica.
- Eurobot 2012 "Treasure island" La Ferté-Bernard (Francia). Robots Crispín y Automático (Eurobotics Engineering-UAH). Apoyo técnico fabricación mecánica.
- Eurobot 2014 "Prehistobot" Dresden (Alemania). Robots Grosnik y Seskäpa (Eurobotics Engineering-UAH). Diseño y fabricación mecánica.

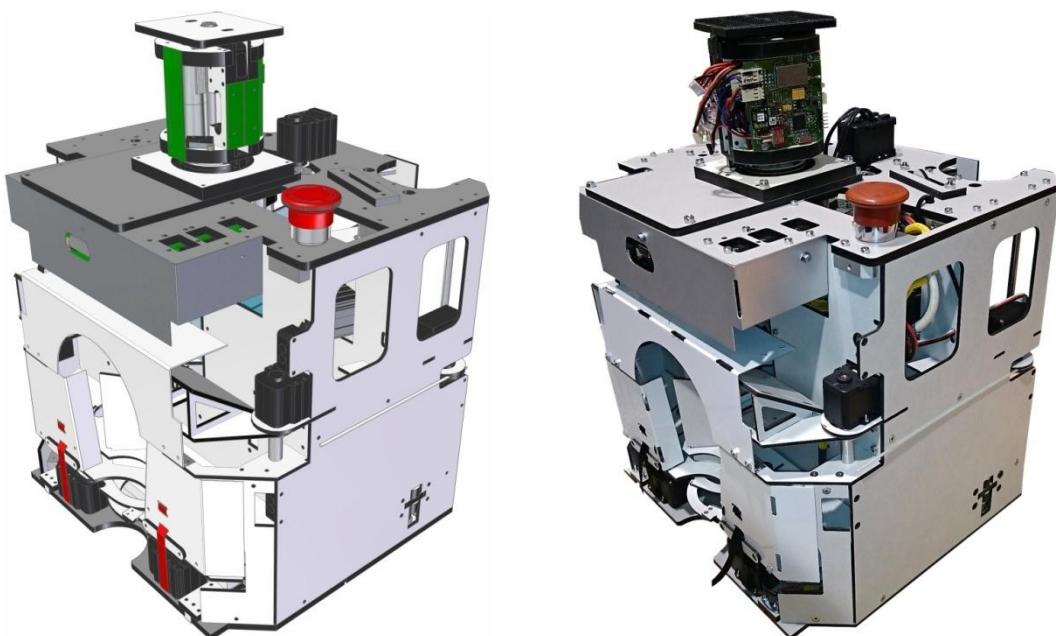


Figura 1-2 Robot "Petinto" vista trasera Diseño 3D a la izquierda, Fotografía a la derecha

La prueba de Eurobot se ha jugado sobre el terreno de juego de la Figura 1-3.

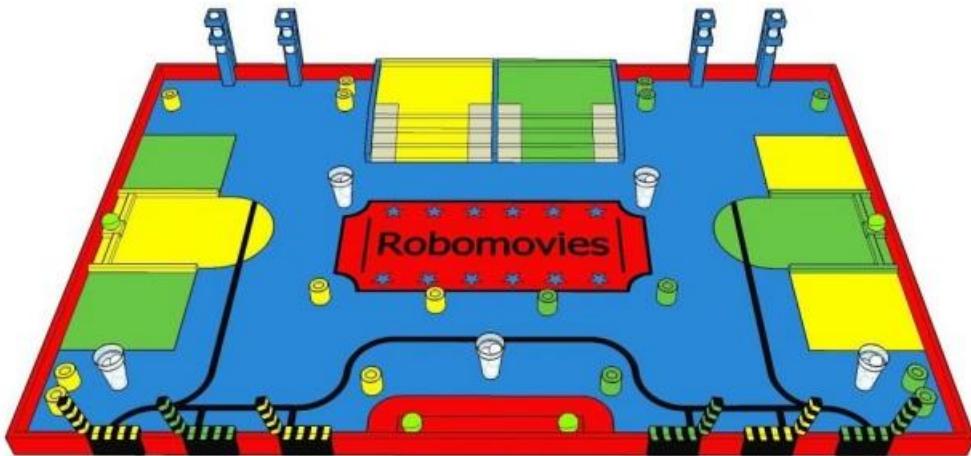


Figura 1-3 *Campo Eurobot 2015, con elementos de juego* (Eurobot, 2014)

En la prueba de Eurobot 2015 se plantea la temática “Robomovies”, y se crean un gran número de elementos de juego relacionados con el mundo del cine. Para comprender la normativa es preciso explicar cada uno de los elementos por separado.

- Los “stands” son cilindros de madera de 60 mm de diámetro y 70 mm de longitud. Se encuentran colocados por el campo en posiciones conocidas, siendo el objetivo recolectar los de tu color y almacenarlos en unas áreas del campo conocidas como “áreas de construcción”.  
Existe un área común para los dos equipos elevada 22 mm sobre el piso y otra a la que solo pueden acceder los robots de cada uno de los equipos. Esta última corresponde con el área de salida de los robots del equipo.
- Las “bombillas” son pelotas de tenis. Existen dos pelotas colocadas en posiciones conocidas del campo, y otra por cada equipo colocada en el área de salida de los robots. Cada uno de los equipos tiene la posibilidad de que uno de sus robots comience el partido con una de las pelotas de la zona común en su interior.  
El objetivo es colocar estas pelotas sobre los stands en las zonas de construcción. Los stand con bombilla formaran un foco y tendrán una mayor puntuación. Debido a que el numero de stands del color de cada equipo es 8 y el máximo número de pelotas recolectables por cada equipo es de tres o dos dependiendo de si los robots oponentes parten con su pelota de la zona común dentro de uno de sus robots, para alcanzar la máxima puntuación será necesario apilar stands, colocando la bombilla sobre el stand superior.
- Las “palomitas” son pelotas de poliestireno expandido de 40 mm de diámetro. Es posible conseguirlas de dos elementos contenedores diferentes presentes en el campo:

- Las “máquinas de palomitas” son dispensadores formados por un tubo de PVC con un diámetro interno de 44 mm colocado verticalmente y del cual es posible extraer las palomitas de una en una. Estas máquinas tienen cargadas 5 palomitas y hay en total 4 máquinas colocadas en posiciones conocidas del campo.
- Las “cestas de palomitas” son 5 vasos distribuidos por el campo en posiciones conocidas, teniendo cada uno cuatro palomitas en su interior.

El objetivo es recolectar el máximo número de palomitas y almacenarlas en diferentes áreas del campo.

Existen tres zonas “los cines” para cada equipo donde es posible almacenar en cada uno de ellos un solo vaso con todas las pelotas que pueda contener en su interior. También existen dos “canastas de palomitas” en una zona protegida para cada equipo donde es posible colocar un número limitado de palomitas.

- Las “claquetas”, son una simulación de 3 claquetas de cine para cada equipo colocadas en posiciones conocidas. El objetivo es cerrarlas.
- Las “escaleras” son cuatro escalones de 22 mm de altura de contrahuella y 70 mm de profundidad de huella. El objetivo es que el robot sea capaz de subir estos escalones hasta la repisa superior.
- Las “alfombras rojas” son tiras de tela que el robot lleva precargadas y tiene que desplegar sobre los cuatro escalones de la escalera en dos posiciones conocidas.

Cada equipo estará formado por un máximo de dos robots, teniendo cada uno una limitación diferente de tamaño en estado contraído y desplegado siempre después del arranque.

Por último es importante conocer que cada partido tendrá una duración de 90 segundos.

---

## 1.4. Problema a resolver y objetivos

Aunque la normativa propone gran cantidad de acciones puntuables, cada equipo puede elegir una estrategia de juego y una serie de elementos limitados por los que competir según esta. Para poder competir y homologar un robot basta con demostrar que es capaz de ganar un partido sin oponente.

La experiencia en esta competición de "Eurobotics Engineering" les ha demostrado que para competir al máximo nivel los robots tienen que tener capacidad de jugar por la puntuación máxima, manipulando todos los elementos de juego propuestos.

Es por esto que se han diseñado dos robots, el número máximo por equipo, y se han dotado a estos de sistemas para manipular todos los elementos de juego repartiendo las tareas en algunos casos y en otros, dotando a ambos robots de las mismas capacidades para potenciar la versatilidad en la estrategia durante el juego.

Todas estas decisiones han sido tomadas en la fase previa al trabajo de este proyecto, con la colaboración y aportaciones de todos los integrantes del equipo. En esta fase se ha creado la división de tareas del Diagrama 1-1.

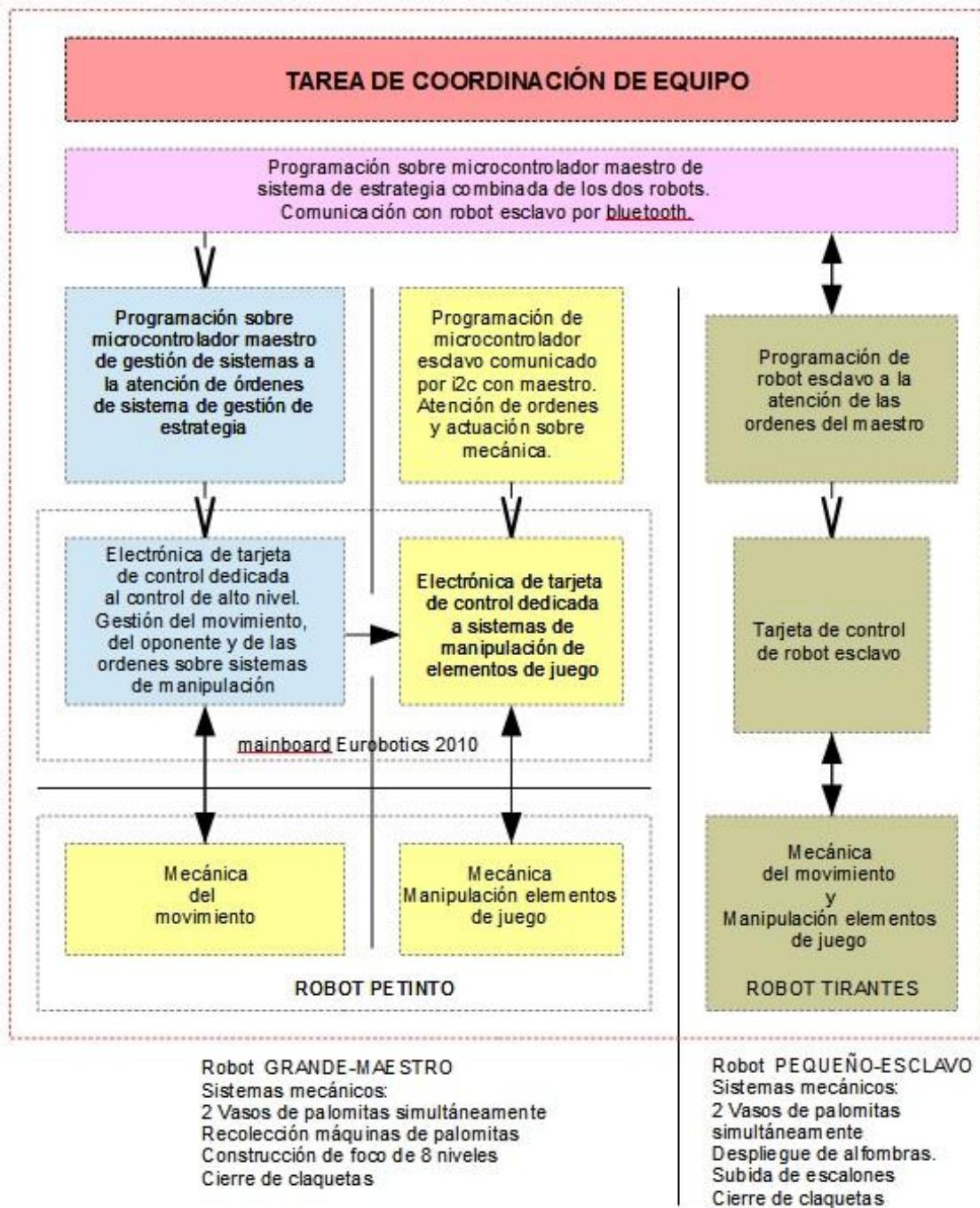


Diagrama 1-1 *Diagrama organización tareas equipo Eurobot 2015.*

En este esquema cada color corresponde a una agrupación de tareas asignada a un integrante del equipo. Con la excepción de las tareas de color magenta que cuentan con dos integrantes para su desarrollo.

El trabajo documentado en este proyecto corresponde a las tareas marcadas de color amarillo, que como se puede apreciar es una parte del desarrollo global perfectamente acotada del resto de los sistemas del proyecto. Esta contiene por orden de desarrollo las siguientes fases:

- Diseño y construcción mecánica del robot maestro partiendo de unas especificaciones completamente cerradas para los sistemas de la mecánica del movimiento, que están definidas por la electrónica y el software de control utilizado. También se ha partido de unas especificaciones iniciales para el desarrollo de los sistemas de manipulación de elementos de juego, definidas en base a la estrategia de juego inicialmente marcada, pero no encontrándose completamente cerradas ya que en este punto solo se ha partido de bocetos de sistemas y aproximaciones a la velocidad de actuación de estos.  
En esta fase se ha creado toda la documentación necesaria para la fabricación de los sistemas mecánicas por parte de cualquier empresa especializada según materiales o trabajos necesarios. Se gestionaron los planos generados, creando un sistema de codificación que permite un fácil control y archivo de los mismos.
- Análisis y adaptación de la electrónica dedicada en la tarjeta principal para el control de los diferentes actuadores y sensores de los sistemas de manipulación de elementos de juego.
- Programación del microcontrolador esclavo para el control exclusivo de los sistemas de manipulación de elementos de juego. Previo a esta fase se ha colaborado con el programador del microcontrolador maestro para definir la estructura de datos que fue compartida por el bus i2c para el envío de órdenes de control desde este.
- Generación de manual de usuario para su utilización por el programador de nivel superior al de este sistema y por los integrantes del resto del equipo para la reparación de posibles averías mecánicas.

Este proyecto según se ha planteado no ha pretendido alcanzar un alto nivel de complejidad en ninguna materia concreta, su objetivo ha consistido en abarcar el máximo número de materias y resultar altamente interdisciplinar, tal y como se plantean las Ingenierías Técnicas Industriales.

## 1.5. Estructura de la memoria

La memoria del proyecto está organizada en el mismo orden en el que han sido realizadas las diferentes fases de diseño. Estas por orden son el diseño mecánico con la generación de toda la documentación necesaria para la fabricación, la adaptación de la electrónica compartida con el resto de sistemas y la programación para el control de los sistemas mecánicos diseñados.

Esta parte del documento se encuentra estructurada en 8 capítulos, los cuales se pueden dividir en 3 partes claramente diferencias. En la primera parte formada por los capítulos del 2 al 7 partiendo de unas especificaciones de diseño mecánico, y después de analizar las fases de diseño utilizadas y tras una aproximación al diseño conceptual de las soluciones y su dimensionamiento, se sucede la descripción de las soluciones tomadas para cada uno de los sistemas de manipulación. Se han dividido los principales sistemas en diferentes capítulos, incluyendo los cálculos necesarios para asegurar la consecución de las especificaciones. Se concluye esta primera parte con algunos cálculos necesarios para posteriores tareas de programación influencias por las características físicas del robot.

La segunda parte diferenciada que corresponde al capítulo 8 describe la electrónica de control utilizada y su adaptación a las señales de los diferentes sensores y actuadores utilizados en el diseño mecánico. Por último en el capítulo 9 se aglutina toda la programación realizada utilizando como base las librerías Aversive.

Previa a la realización de las fases descritas en esta memoria existe una etapa inicial colaborativa con el resto del equipo, en la que de forma conjunta se han definido las especificaciones y los tiempos de desarrollo de las diferentes partes en que ha sido dividido el proyecto global.

## **2. Diseño mecánico**

---

Este capítulo se encuentra dividido en tres partes claramente diferenciadas.

En la primera se recogen las especificaciones para el diseño mecánico surgidas en la etapa colaborativa previa con el resto del equipo y en la que con las aportaciones de la experiencia común se definen las líneas a seguir y los objetivos funcionales mínimos a alcanzar. En esta fase se va a definir una estrategia de juego que será la base para la definición de las especificaciones de diseño mecánico y que serán consideradas especificaciones u objetivos de proyecto.

En la segunda parte se resume la metodología que se va a emplear para acometer el diseño mecánico, se trata de un resumen de las diferentes fases, comenzando desde el estudio y la recopilación de las especificaciones para cada sistema hasta la fase de generación de la documentación para la fabricación.

En una última parte se resume quizás el trabajo más determinante de la fase de diseño mecánico, en la que es estudián cada uno de los sistemas a integrar, creando una serie de bocetos en 2 dimensiones que se aproximen con la mayor exactitud posible a las soluciones mecánicas que se emplearan y a partir de estar cubicar y ubicar estas dentro del espacio disponible en el diseño global. En esta fase se localizan cada uno de los sistemas a diseñar dentro del robot y se distribuye el espacio según la importancia y necesidades de cada uno de ellos. Previamente a esta fase lógicamente es necesario definir el volumen del robot en función de las limitaciones de la normativa y de las necesidades derivadas de las especificaciones que pueden influir en este.

### **2.1. Especificaciones de diseño**

---

La mayoría de las especificaciones mecánicas surgen como requerimientos mínimos para la ejecución de una estrategia inicial, siendo esta estrategia inicial la secuencia de operaciones óptima para la consecución de la máxima puntuación sin interferencias de los robots oponentes y que ha sido definida en la fase previa de forma consensuada con el resto del equipo.

La elección de este tipo de estrategia se basa en las características de la prueba planteada, con dos zonas del campo claramente separadas para las operaciones de cada uno de los equipos. La mayor parte de los elementos por los que juega cada equipo se encuentran en su mitad del campo, lo que limita las posibles interferencias entre los dos equipos. Siendo esta situación además en general poco deseada por la mayoría de los equipos de más alto nivel, cuyo interés principal es sumar el mayor número de puntos evitando al máximo la incertidumbre creado en el encuentro directo con robots oponentes.

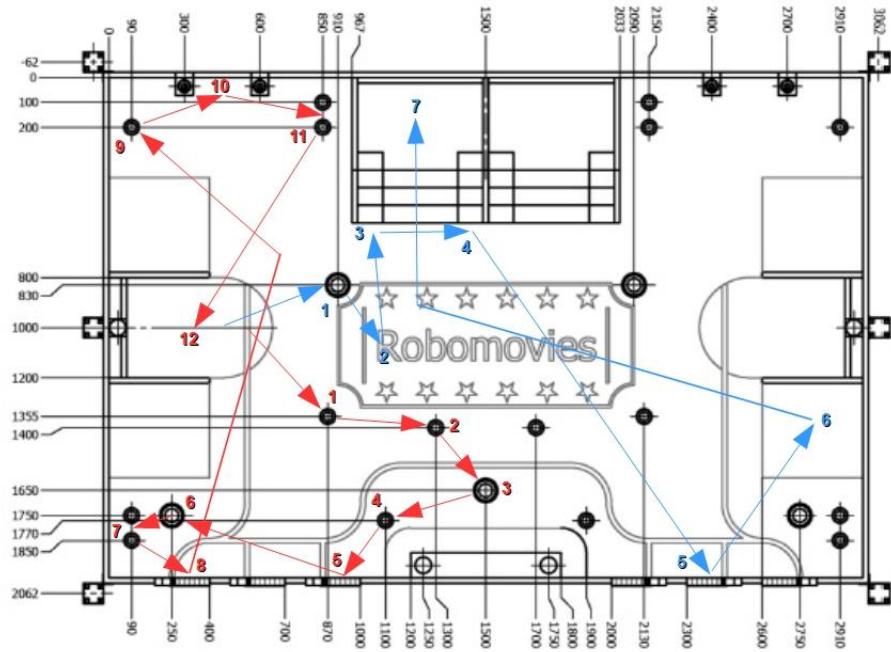


Figura 2-1 Secuencia de movimientos de estrategia optimizada sobre imagen  
(Eurobot, 2014)

Como parte de esta especificación se describe la secuencia de la estrategia inicial que ha sido decisiva para la toma de decisiones de diseño. En la Figura 2-1 se detalla la secuencia de movimientos de esta estrategia optimizada. Por supuesto esta secuencia es alterada en cualquier momento por decisión del sistema de alto nivel de gestión de estrategia ante acciones no esperadas de los robots oponentes.

En color rojo se ha trazado la secuencia optimizada del robot maestro y en color azul la del esclavo, aunque esta última no va a ser analizada en detalle.

Antes de continuar para comprender los movimientos realizados es preciso definir el sistema de locomoción que se va a utilizar en el robot. Este se va a construir con una estructura típica bidireccional con dos ruedas de tracción paralelas siempre que sea posible centradas en el centro geométrico del robot y un encoder por rueda colocado en el mismo eje de estas y en disposición exterior a estas. Tanto las ruedas de tracción como las del encoder se colocarán tan próximas como sea posible a los laterales del robot para mejorar la precisión del sistema odómetrónico utilizado.

A continuación se analiza secuencia a secuencia la trayectoria propuesta detallando las especificaciones necesarias en el diseño:

### Secuencia 1-3

El robot maestro siempre será colocado en la zona de salida en primer lugar, en dirección al vaso de la posición 3. El objetivo de la primera secuencia de movimiento

será alcanzar la posición número 3, donde se encuentra el vaso central, en un movimiento rectilíneo uniformemente acelerado y decelerado, con un límite de velocidad máxima de 1 m/s y con ambas aceleraciones con un límite máximo de 3.6 m/s<sup>2</sup>. Estos valores son estimativos de robots de características similares construidos para anteriores competiciones, en cualquier caso una vez concluido el diseño mecánico serán calculadas en función del centro de masas, su distancia a los apoyos traseros y delanteros, y el coeficiente de rozamiento de las ruedas. Este último será obtenido de forma empírica.

En esta secuencia el robot tiene que ser capaz de manteniendo la trayectoria en línea recta y sin variar la velocidad máxima, recoger los dos stands que se encuentran a los lados de esta trayectoria e inmediatamente y sin interferir ni con el movimiento ni con la próxima acción programada colocarlos internamente en el almacén de stands. De esta situación se deduce que será necesario duplicar el sistema de recogida de stands creando dos paralelos colocados a ambos lados del robot.

Una vez alcanzada la posición número 3, el robot deberá recoger el vaso y llevarlo consigo o almacenar las palomitas en el interior del robot. El sistema de recogida deberá situarse en la parte delantera del robot y una vez alcanzada la posición sujetarlo en un tiempo máximo de 0,2 segundos, que corresponde con el tiempo de giro de un servo Dynamixel AX12 para 60. El objetivo final de esta acción es poder alcanzar este vaso antes que ningún otro oponente, ya que para llegar en línea recta hasta él, es preciso recoger los stands que se encuentran en la trayectoria, y ningún equipo puede permitirse perder estos puntos que con la construcción de un foco tienen mayor puntuación que el vaso objetivo. Para nuestra estrategia los puntos de este vaso supondrán la diferencia con un equipo de similar estrategia a la nuestra.

### **Secuencia 3-4**

En esta secuencia el robot girará sobre sí mismo hacia la posición número 4 y se desplazara hasta esta con la velocidad y aceleración definida para la secuencia anterior. El giro se controlara como un movimiento en direcciones opuestas de ambas ruedas con las mismas limitaciones de velocidad y aceleración en cada una de ellas. Como la velocidad angular dependerá de la distancia final entre las ruedas del diseño se va a estimar inicialmente una velocidad angular de 300 grados/seg y una aceleración y deceleración constante de 1500 grados/seg<sup>2</sup>.

El objetivo final de definir las velocidades de los movimientos de traslación y rotación, es poder calcular la velocidad de maniobra de aquellos sistemas que deban operar de forma paralela al movimiento, ya que una vez alcanzado un elemento de juego este deberá ser procesado y el sistema tendrá que quedar en posición de recibir un nuevo elemento de ese tipo, antes de alcanzar la próxima posición de recolección.

En el caso del sistema de recolección de stands la velocidad de maniobra será especialmente crítica y quedara definida por el tiempo de recolección mínimo entre dos stands, que deban ser recolectados por el mismo sistema de almacenaje del robot. Como ya se ha adelantado en la secuencia 1-3 el robot tendrá dos sistemas de almacenaje paralelos y con funcionamiento completamente independiente.

Estudiando la secuencia de movimiento inicial y se puede comprobar fácilmente que el tiempo mínimo entre recolecciones en el mismo almacén se dará entre las posiciones 2 y 4, ocurriendo ambas en el almacén izquierdo. Será necesario calcular

dicho tiempo con las velocidades indicadas y utilizarlo como objetivo de velocidad de procesado de este sistema.

### **Secuencia 4-5**

En la posición número 5 se cerrará la claqueta mediante un sistema de funcionamiento simétrico respecto al eje longitudinal del robot, o con un sistema doble. Esta es una característica de la mayoría de los sistemas a diseñar ya que los robots dependiendo del lado del campo por el que jueguen tendrán que adaptar su programación para realizar un funcionamiento completamente simétrico. Esto resulta inmediato para sistemas de funcionamiento simétrico o sistemas dobles simétricos respecto al eje longitudinal del robot. Los dos sistemas ya comentados pertenecen a ambos tipos de sistemas. Por un lado el sistema de captura del vaso delantero salvo imposibilidad se diseñara en el eje longitudinal del robot y los dos almacenes de stands serán dos sistemas dobles simétricos respecto al eje longitudinal del robot.

No se crea una especificación de tiempo mínima pero se primara que sea un sistema que permita cerrar las claquetas sin detener el movimiento de traslación.

### **Secuencia 5-6**

En la posición número 6 se cogerá el vaso. Este vaso será transportado durante el resto del partido para posteriormente dejarlo en la zona protegida de salida. Se primaran las siguientes características:

- Sistema con amplio margen de error de posicionamiento.
- Sujeción robusta y preferiblemente alojada en el cuerpo del robot para evitar obstaculizar los giros y que otros robots pueden chocar si se encuentra en una posición prominente.
- Se deberá prever un sistema para recuperar las palomitas del vaso de la posición 3 e introducirlas en este vaso.

### **Secuencia 6-7**

La posición 7 está ocupada por dos stands colocados a poca distancia de forma paralela. Los sistemas de recolección de stands de los dos almacenes funcionando de forma independiente y paralela deberán ser capaces de recolectar ambos en el tiempo de una única maniobra. Estos mecanismos deberán disponer de un sistema para atrapar los stands que permita iniciar la marcha atrás o giro inmediatamente después de alcanzar la posición de recolección. El tiempo de maniobra de bloqueo no debería superar los 0,3 segundos, que corresponde con el tiempo de giro de un servo Dynamixel AX12 para 90 grados.

### **Secuencia 7-8**

Se espera el mismo funcionamiento que para la secuencia 4-5.

### **Secuencia 8-9**

Se espera el mismo funcionamiento que para la secuencia 3-4.

### **Secuencia 9-10**

El sistema de recolección de palomitas deberá ser capaz de vaciar las palomitas de los dos dispensadores paralelos, por lo tanto será necesario diseñar un sistema doble simétrico. El diseño del robot en esta posición deberá permitir referencia la posición de los encoder del sistema de odometría chocando con la pared o base de las máquinas dispensadoras. Para este sistema no se establece un límite de tiempo mínimo de operación, pero si se deberá asegurar un porcentaje de error mínimo y una alta tolerancia al error de posicionamiento en el eje X (longitudinal del campo).

### **Secuencia 10-11**

Se espera el mismo funcionamiento que para la secuencia 6-7.

### **Secuencia 11-12**

La posición 12 corresponde con la zona de salida y la zona de construcción protegida en la que no tendrán acceso permitido los robots oponentes.

Se deberá diseñar un sistema capaz de vaciar las palomitas recolectadas en las máquinas de palomitas en las canastas de esta zona.

En esta zona también se dejara el vaso de la zona 6 con las palomitas del vaso de la zona 3 en su interior.

Por último el robot será capaz de construir una torre apilando 8 stands con una pelota de tenis en la parte superior, que lleva precargada antes del inicio del partido. Para reducir el tiempo de construcción se plantea como solución el almacenaje previo en forma de torre. Cada uno de los sistemas de captura y almacenaje de stands podrán almacenar previamente una altura de cuatro stands para no superar la limitación de altura durante el juego. Esta altura lógicamente podrá ser superada en las zonas de construcción durante la elevación del foco.

## **2.2. Fases de diseño mecánico**

Partiendo de las especificaciones previas se ha estructurado el proceso de diseño en una serie de fases sucesivas que han permitido acometer un proyecto de este tipo, con la complejidad intrínseca de un sistema con una alta densidad de subsistemas, en un espacio muy reducido y con una alta interdependencia entre los mismos.

Las fases principales de esta estructura se describen a continuación.

### **Fase 0. Estudio de las especificaciones de diseño**

En esta fase se van a plantear las especificaciones mecánicas con los requerimientos mínimos para el desarrollo de la estrategia inicial y van a ser analizadas en conjunto. La estrategia inicial es la secuencia de operaciones óptimas para la consecución de la máxima puntuación. Esta variará en función de las acciones de los robots del equipo oponente. Esta fase ha sido desarrollada en el apartado 2.1.

### **Fase 1. Estudio del espacio disponible para los sistemas**

Previo al análisis de cualquier sistema de forma individual, en un desarrollo de esta tipología y con un elevado número de sistemas en un espacio muy reducido, es preciso un análisis en conjunto de todos los sistemas. En esta fase se comenzara a definir el funcionamiento de los sistemas de forma esquemática, siendo su objetivo acotar de una manera aproximada el volumen disponible para cada sistema y por supuesto su disposición dentro del volumen total.

También se presta especial atención en la distribución de masas para conseguir en el diseño global un centro de gravedad lo más bajo posible. La aceleración máxima del robot es un punto muy importante en el diseño.

### **Fase 2. Diseño de estructura portante**

Aunque esta fase se concluirá en la última fase de desarrollo una vez diseñados todos los sistemas, es necesario antes de comenzar el diseño de estos, tener una primera aproximación al “esqueleto” que los contendrán.

Se planteará un primer diseño esquemático de la estructura portante, que deberá tener la solidez necesaria.

### **Fase 3. Diseño de los sistemas**

Esta fase se divide en múltiples fases que pueden repetirse de manera recursiva y para cada uno de los sistemas que han de diseñarse.

Se partirá de unas necesidades u objetivos perfectamente definidos, de un volumen máximo modificable según necesidades y de un primer diseño esquemático, creado en la fase previa.

Las fases serán las siguientes:

1. Diseño funcional del sistema. No se incluirán sistemas de accionamiento aunque si se tendrá en cuenta el posible espacio necesario.
2. Creación de prototipo funcional rápido, si se cree necesario. Se fabricara con materiales de fácil manipulación, como papel, cartón o madera y su objetivo será comprobar el funcionamiento de la idea y su interacción con los elementos de juego. Según los resultados y su evaluación se retrocederá a la fase previa o se continuara con el diseño.
3. Dibujo 2D en programa de CAD sin incluir sistemas de accionamiento.
4. Estudio de los sistemas de accionamiento, servomotores, motores, sistemas de transmisión, etc. Se dimensionaran mediante su análisis físico, calculando par, velocidad, etc. para conseguir las especificaciones requeridas de los sistemas de manipulación.
5. Conclusión de diseño 2D con sistemas de accionamiento y dibujo 3D del sistema completo. Una vez concluido cada sistema se ensamblara como conjunto en el diseño global.
6. Análisis del sistema mediante simulación de articulación de movimientos en diseño 3D, analizando interferencias entre piezas.

#### **Fase 4. Análisis de diseño global**

Una vez concluidos todos los sistemas y ensamblados en el diseño 3D global, se finalizara el diseño de la estructura portante, incluyendo todas las fijaciones necesarias. También se definirá el trazado del cableado, practicando las aberturas necesarias a través de los sistemas diseñados y de la estructura.

Se analizaran las posibles interferencias en los movimientos entre los distintos sistemas entre sí y con la estructura, y también las posibles entre la tornillería y todos los demás elementos.

Se concluirá el diseño realizando los siguientes estudios y reportes:

- Listado de piezas mecanizables indicando cantidad, material y método de fabricación.
- Listado de tornillería por unión de tipo de pieza indicando cantidad, métrica, longitud y DIN.
- Listado de sensores, indicando cantidad y tipo.
- Listado de motores o servomotores, indicando cantidad y tipo.
- Listado del resto de piezas (poleas, correas, engranajes, rollones, etc.).
- Peso del robot completo y posición de centro de masas.

#### **Fase 5. Generación de documentación para fabricación**

Se prepararan los planos constructivos de todas las piezas con el acotado necesario según norma ISO, indicando tolerancias necesarias según pieza. Se creara sistema de codificación para el control y archivo de los planos generados.

La realización de cada una de estas fases ha sido documentada en cada uno de los subapartados siguientes. En general solo se muestran en ellos el resultado final del diseño y el razonamiento empleado para su consecución, obviando las opciones descartadas para cada diseño.

### **2.3. Análisis y dimensionado de sistemas**

En esta fase inicial del desarrollo se ha creado una aproximación a la disposición de todos los sistemas que van a conformar el conjunto del robot. Aunque en este apartado solo se va a documentar el resultado final se trata de una fase muy amplia, en la que partiendo de las especificaciones de entrada se ha definido el funcionamiento y la disposición de cada sistema dentro del conjunto al menos de forma esquemática, y sin necesidad de definir los actuadores que producirán el movimiento. La razón de no profundizar en el desarrollo de cada uno de los sistemas en esta etapa es poder agrupar en su apartado correspondiente el desarrollo integral utilizado.

El objetivo final de esta fase ha sido acotar de una manera aproximada el volumen disponible para cada sistema y por supuesto su disposición dentro del volumen total.

Como punto de partida se ha definido la forma de la planta del robot. Esta aunque no tiene forma definida por la normativa si tiene un perímetro máximo de 1200 mm con todos los sistemas replegados antes del inicio del partido. Las formas típicas de los robots participantes en esta prueba van desde formas circulares perfectas, pasando por formas poligonales, hasta rectangulares o cuadradas. Aunque el diseño habrá de adaptarse a la funcionalidad de los sistemas del robot la forma circular es la que más ventajas tiene de cara al movimiento de giro de un robot bidireccional. Un robot con forma circular puede girar sin chocar con el oponente o las paredes, no pudiendo quedar bloqueado en ninguna situación.

En nuestro diseño esta forma está condicionada por dos de las especificaciones. El robot moviéndose en línea recta desde la zona de salida hasta el vaso de la posición número 3 deberá ser capaz de recoger los dos stands a ambos lados en las posiciones 1 y 2, esta circunstancia junto con la necesidad de diseñar en la parte trasera un sistema que sea capaz de vaciar las dos máquinas de palomitas al mismo tiempo colocándose entre ambas, han definido una planta rectangular con la mayor longitud en dirección perpendicular a la dirección de avance. Como desde el punto de vista razonado en el párrafo anterior para facilitar el giro es la peor situación posible se han achaflanado las esquinas al máximo para mejorar esta capacidad.

Una ventaja de este diseño, teniendo en cuenta las especificaciones es el mayor distanciamiento posible tanto entre las ruedas de tracción como entre los encoder.

Voy a describir de forma esquemática cada uno de los sistemas y su disposición dentro del conjunto, comenzando por los sistemas que han condicionado el volumen y la disposición de otros menos críticos. En la Figura 2-2 y Figura 2-4 se muestra la disposición y el volumen de los sistemas generados en esta etapa.

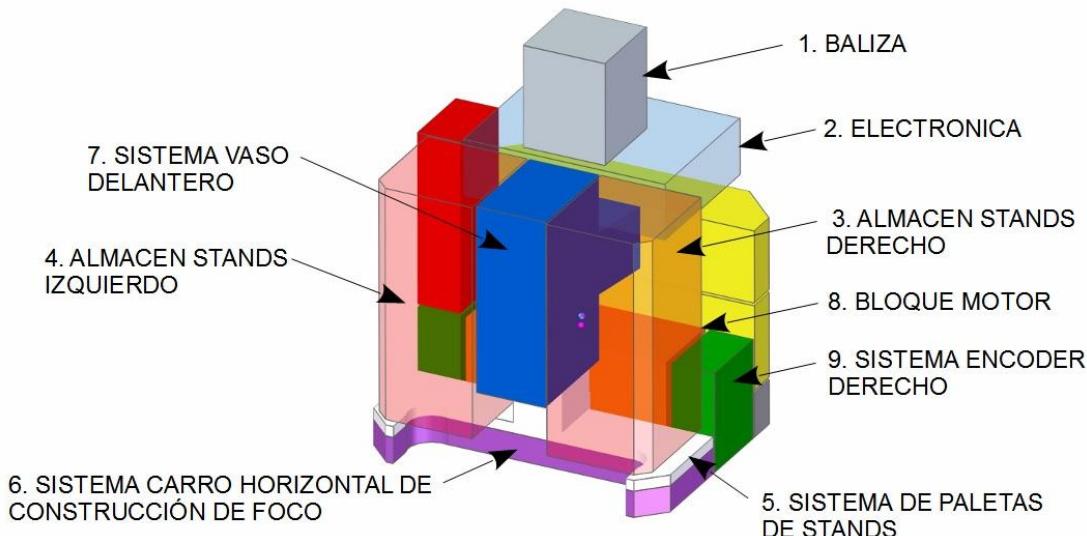


Figura 2-2 Conjunto Bloques Vista Delantera

Bloque del sistema locomotor (8). Este bloque se ha representado de color naranja.

Se ha diseñado un sistema compacto que bajo una de las especificaciones ha sido colocado lo más próximo posible al centro geométrico del robot, encontrándose ligeramente desplazado hacia la parte trasera por el mayor tamaño del sistema de manipulación de stands situado en la parte delantera. Esta disposición descentrada como se calculara en los parámetros de finalización del diseño creara una limitación de aceleración máxima diferente en función de la dirección del avance.

Este bloque junto con los sistemas de los encoder (9 y 10) de odometría que están colocados a sus lados divide el robot por la mitad creando una separación entre sistemas. Los bloques de los encoder según la especificación han sido colocados en el mismo eje que el sistema de tracción.

El sistema de manipulación de stands está compuesto por cuatro sistemas independientes que trabajan de forma conjunta. Estos sistemas aparecen en la Figura 2-2 con las numeraciones 3, 4, 5 y 6.

El punto de partida en este diseño ha sido la especificación de la secuencia 1-3 en la que es necesario recoger dos stands a ambos lados de dicha trayectoria. Se ha comprobado mediante el dibujo en dos dimensiones Figura 2-3 que la máxima distancia de ambos stands al centro de la trayectoria es de 112 mm para el de la posición 1 y 137 mm para el de la 2. En base a estas distancias se ha ajustado en 260 mm la abertura máxima de la “boca” donde van a ser alojados los stands. Aunque esta distancia no permite la recepción directa del segundo stand, se ha comprobado empíricamente que el margen de recepción para elementos redondos aumenta en 1/3 del diámetro de dicho elemento por cada lado de la abertura, siendo de 20 mm más para el caso de los stands con un diámetro de 60 mm. Y finalmente con un margen de 130+20 mm se puede asegurar la recepción correcta del segundo stand a 137 mm de la trayectoria.

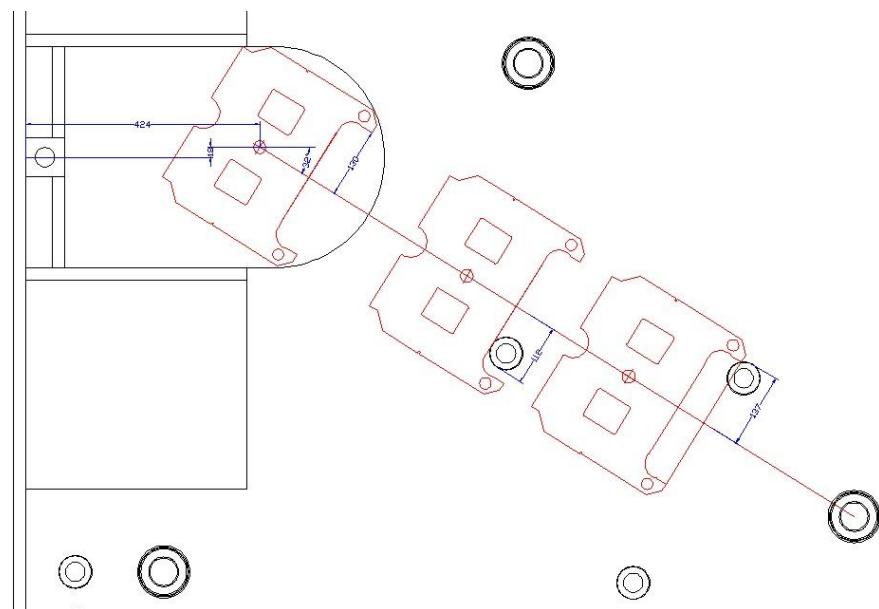


Figura 2-3 Secuencia 1-3

Asegurada la entrada de los stands por el propio movimiento de avance en un margen muy amplio, se ha creado un sistema denominado de “paletas de stands”, que corresponde al bloque número 5 de color blanco, que dispone de dos paletas articuladas por un eje vertical y situado en el eje longitudinal de movimiento del robot. Por su disposición se trata de un sistema simétrico, pudiendo cada paleta girar 180 grados sin interferir con la otra paleta. Estas paletas en posición de captura de stands se colocan orientadas en la dirección de avance y una vez que es detectada la entrada de un stand, una de las paletas se cierra contra él inmediatamente. El sistema será lo suficientemente rápido para evitar que los stands puedan salir despedidos en el choque contra el robot. Este sistema doble permite bloquear los stands en una posición fija exacta para que el sistema de almacenaje pueda cogerlos con precisión y sin detener el movimiento del robot.

El sistema de almacenaje está formado por una pinza que es capaz de sujetar los stands y elevarlos 80 mm de altura verticalmente desde la posición exacta en que son bloqueados por las paletas de stand contra un ángulo de 90 grados. El funcionamiento de almacenaje consiste en elevarlos por encima de la altura de un stand que es de 70 mm, para permitir que otro pueda ser introducido debajo de este. Ocurrido esto la pinza suelta el stand que sujetaba cayendo este sobre el inferior, y desciende para elevar el nuevo stand capturado, junto al que reposa ahora encima. Este ciclo se repetirá hasta la captura del cuarto stand por almacén que no será elevado ya que se superaría la altura máxima del robot durante la fase de recolección. Este último stand permanecerá bloqueado por la paleta de captura.

Este proceso se repetirá por los dos almacenes hasta la captura de los 8 stands disponibles.

Como último sistema se ha ideado un “carro horizontal” para en el proceso de construcción del foco de ocho alturas desplazar los stands desde la torre derecha a la izquierda o viceversa. Consiste en una paleta que se desplaza horizontalmente a lo ancho de todo el robot. El espacio para el desarrollo de este sistema corresponde al bloque número 6 de color morado.

El sistema de captura del vaso delantero corresponde al bloque número 7 de color azul. Este sistema será capaz de coger el vaso con una pinza por su borde superior y levantarla mediante un movimiento giratorio hasta que las pelotas caigan en el interior del robot a través de una rampa hasta el vaso trasero. Será necesario prever un sistema que evite que las pelotas caigan desde la rampa hasta que no se haya capturado el vaso trasero, que corresponderá al de la posición 6.

En la parte trasera del robot se ha concentrado todo el sistema de manipulación de palomitas con la excepción de la pinza delantera.

Comenzando desde abajo el bloque número 13 está reservado para alojar un sistema formado por dos pinzas que capturarán el vaso trasero. Para facilitar el diseño de este sistema se ha optado por la utilización de dos servos, uno por pinza con movimientos sincronizados.

Este sistema con dos servos también va a permitir utilizar parte del movimiento de cada uno de ellos en su recorrido final para provocar la apertura y cierre de las puertas de descarga

de palomitas. Este almacén se encuentra inmediatamente encima de este piso en el bloque número 12 de color amarillo.

En el bloque número 11 se va a implementar un sencillo sistema de vaciado de las máquinas de palomitas a través de dos rampas simétricas que se introducirán empujando la primera pelota y conducirán el resto en la dirección de salida de forma continua. En este sistema será necesaria una tercera estructura articulada en forma de bandeja para recoger e introducir las pelotas hacia los almacenes colocados en el nivel inferior.

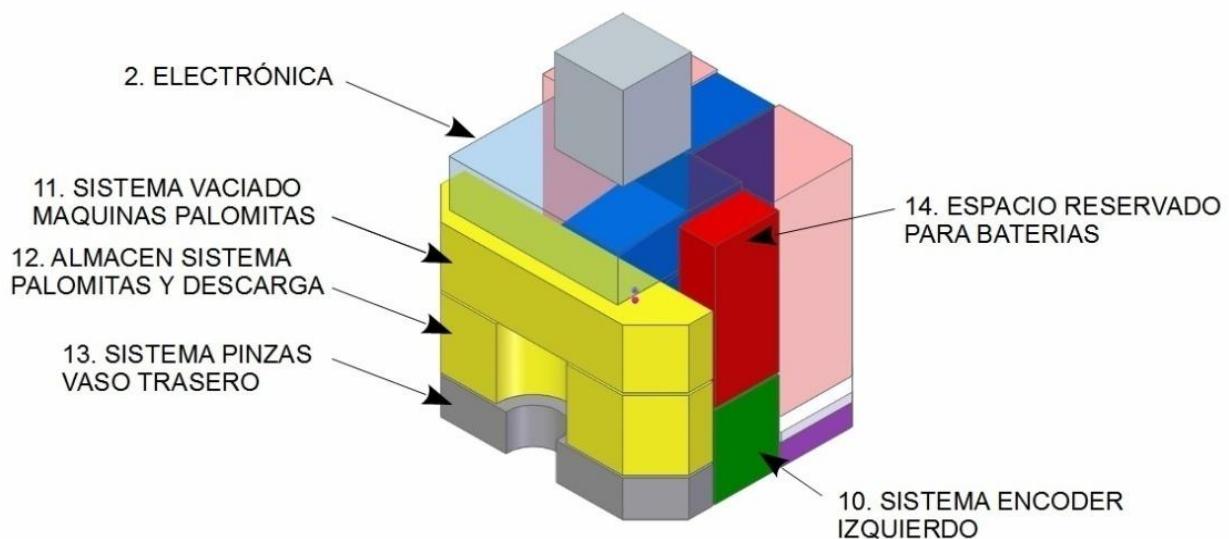


Figura 2-4 Conjunto Bloques Vista Trasera

El espacio para la electrónica (bloque número 2, azul transparente) se ha desplazado hasta el nivel superior para facilitar su accesibilidad. Su posición a la máxima altura también ayuda a bajar el centro de masas del conjunto final.

Por último el espacio disponible encima de los sistemas de los encoder ha sido reservado para el alojamiento de las baterías. Es importante en un diseño de este tipo orientado a la competición facilitar el cambio rápido y sencillo.



## **3. Estructura portante**

---

Este capítulo se encuentra dividido en dos partes claramente diferenciadas, una primera en la que se detalla el diseño realizado para la estructura que soporta todos los sistemas del robot, en función de unas consideraciones previas basadas en la experiencia adquirida en las anteriores participaciones en esta prueba, y una segunda en la que se detallan los diseños desarrollados para el sistema de tracción y el de control de posición por odometría. Los cuales han sido incluidos en el diseño de la estructura debido a la alta interdependencia entre estos sistemas en general y más especialmente en este caso en particular en el que la estructura está diseñada en torno al bloque motor de tracción.

### **3.1. Cuestiones de diseño previas**

---

Tanto esta fase como las posteriores de diseño del resto de sistemas van a estar influenciadas por los materiales y medios de fabricación a los que se van a tener acceso durante la fase de fabricación. Aunque estos son amplios, yendo desde fresado y torneado de materiales de todo tipo, hasta el corte por láser o labores de chapistería, se van a primar aquellos procesos productivos más adecuados para el desarrollo de prototipos rápidos de bajo coste. Evitándose en lo posible diseño de piezas por arranque de viruta en materiales ferrosos o aluminio.

Para el grueso de las piezas se ha optado por el mecanizado sobre materiales fenólicos de bajo espesor. Aunque el mecanizado en comparación con el corte por láser en términos generales tiene un coste claramente superior y actualmente tan solo se justifica el mecanizado en aquellas piezas que requieren una precisión de ensamblado mecánico, el material elegido con unos tiempos de mecanizado que se aproximan a los del corte por laser va a permitir realizar cajeados a diferentes profundidades y en los casos necesarios ajustes con la precisión suficiente para el alojamiento de pistas de rodamientos y ejes calibrados.

Por otro lado estos materiales compuestos por un laminado a base de hojas de celulosa impregnadas con resinas fenólicas prensadas a alta presión y temperatura, tienen una densidad media de 1.35 gramos/cm<sup>3</sup> que es la mitad que un aluminio 5083 que alcanza los 2.66 gramos/cm<sup>3</sup>. Teniendo en cuenta que el espesor mínimo de las planchas de estos materiales es de 3 mm para conseguir piezas en aluminio con el mismo peso estas no deberían superar 1.5 mm de espesor, y en estos espesores la calidad del aluminio empleado en la empresas de corte por laser no alcanza la rigidez que proporcionan las planchas fenólicas de 3 mm.

### **3.2. Estructura principal y sistema de ensamblado por pestañas**

---

En esta fase del diseño conocida la disposición aproximada de los sistemas definida en el apartado 2.3 se ha definido el tipo y composición de la estructura. Para esta fase de diseño se han tenido en cuenta las siguientes consideraciones:

- Facilitar la accesibilidad a todos los sistemas.
- Prever las operaciones de reparación y sustitución de elementos con propensión a la avería o rotura como pueden ser motores, servos, piezas frágiles, etc.
- El robot debe presentar todas sus estructuras con superficies opacas, fácilmente detectables por los sensores de robots oponentes.
- Como parte de la estructura y siempre que el sistema de tracción esté preparado (tenga potencia suficiente), como va a ser este caso se incluirán lastres lo más próximos posibles a la ruedas, para bajar el centro de masas y optimizar la aceleración máxima.
- La estructura será aligerada a medida que se asciende de cota. En caso de planchas continuas se les realizaran vacíados de aligeramientos, pero siempre sin comprometer el punto número 3 y evitando que su ubicación pueda propiciar el enganche en roces con otros robots oponentes o elementos del campo de juego.
- En los sistemas de tracción bidireccional se colocaran los apoyos delanteros y traseros lo más alejados posibles del eje de las ruedas, que como se explicara más adelante y junto con la disposición del centro de masas son las principales variables para la determinación de la aceleración máxima.
- La estructura debe disponer de unos asideros suficientemente resistentes para soportar el peso del robot, que puede ser importante.

Teniendo en cuenta estas consideraciones y partiendo de la disposición de los sistemas de juego prediseñados se ha ideado una estructura que comparte las características de una carrocería autoportante pero sin comprometer la accesibilidad tal y como sucede en un coche actual, siendo parte del conjunto del carenado exterior el que soporta la mecánica interna.

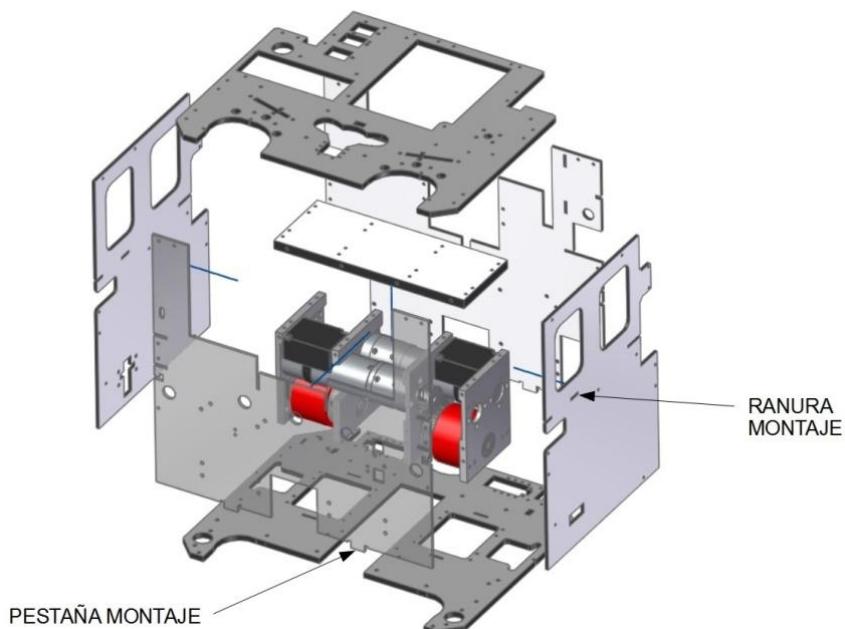


Figura 3-1 *Perspectiva estructura EXPLOSION*

Esta estructura se ha construido en torno al bloque motor, obteniendo este su rigidez del conjunto de la estructura y a su vez siendo este bloque una parte decisiva de la propia rigidez de la estructura. Es la unión del conjunto de piezas con sus diferentes uniones perpendiculares la que conforma una estructura altamente estable y rígida.

Las uniones entre piezas se han realizado en función del tipo de material y grosores a unir. Todas las piezas unidas al bloque motor que está formado por piezas de aluminio de 8 mm de espesor han sido firmemente atornilladas a los cantos de estas piezas. La tapa del bloque motor de material fenólico de 10 mm también dispone de taladros roscados en sus cantos.

Para el resto de piezas se ha creado un sistema de ensamblado altamente estable, mediante la creación de pestañas en los contornos de las piezas que encajan en ranuras creadas en los encuentros perpendiculares con otras piezas, ver Figura 3-1. El bloqueo final de las piezas se consigue mediante unas pequeñas piezas de aluminio con forma de cubo alargado con dos taladros roscados en una cara y otro centrado entre estos en la cara contigua a 90 grados, a modo de escuadras.

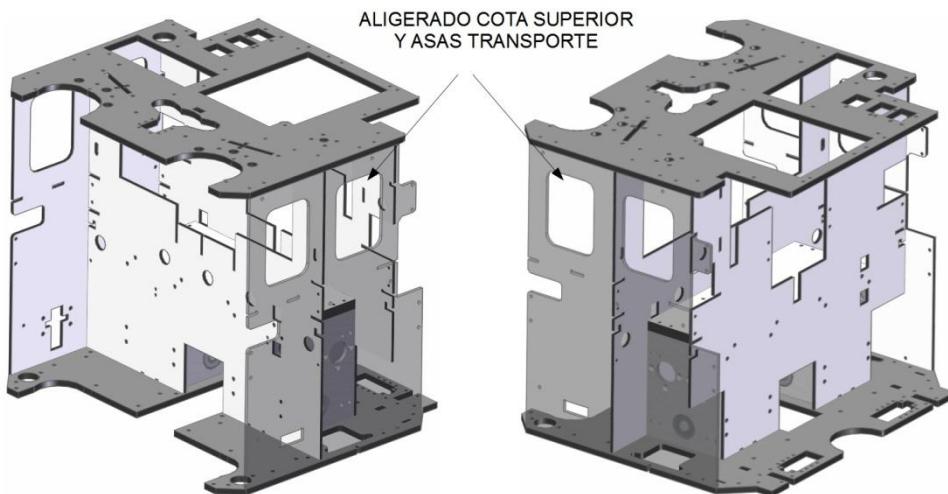


Figura 3-2 Perspectiva estructura izquierda delantera, derecha trasera.

La estructura creada va a permitir acceder a todos los sistemas ya que estos no van a quedar encerrados por ella, sino que serán accesibles por la parte delantera y trasera que quedara abierta. El acceso lateral también será fácilmente practicable ya que las dos tapas laterales podrán ser retiradas mediante 5 tornillos de cabeza Allen.

Como se ve en las imágenes la estructura ha sido aligerada desde media altura hasta la cota superior. Las aberturas superiores de las tapas laterales cumplen una segunda función, pudiendo ser utilizadas como asas de manipulación y transporte. Ver Figura 3-2.

En la base del robot se ha creado una abertura longitudinal para permitir retirar con facilidad todo el bloque de paletas de stands, ya que de otro modo quedaría en una posición muy comprometida en caso de avería. El sistema diseñado va a permitir retirarlo quitando poco más de cuatro tornillos, descolgándolo hacia abajo. Ver Figura 3-3.

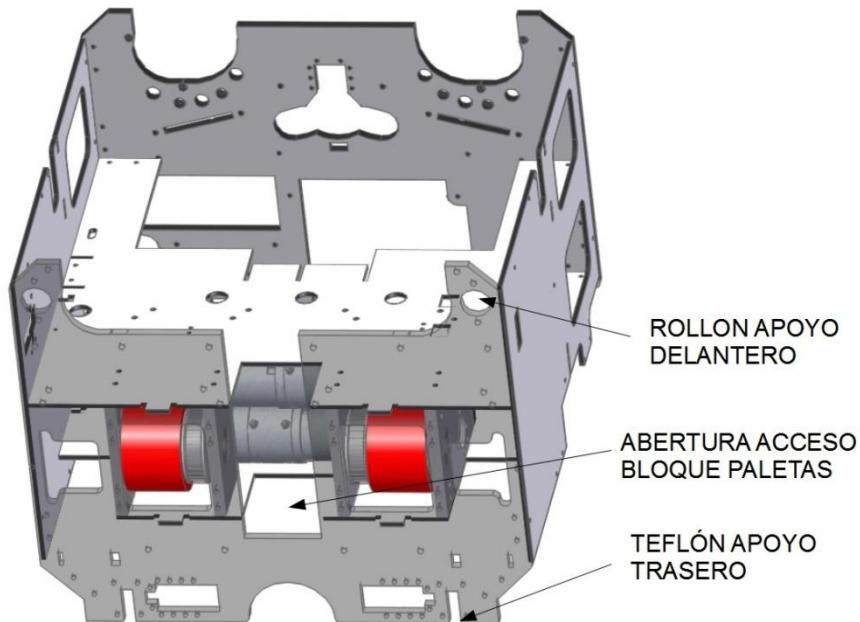


Figura 3-3 Perspectiva estructura inferior

La base del robot ha sido dotada de tres conjuntos de lastre, dos en la parte delantera, en parte del espacio reservado para el sistema del carro horizontal y otro en la parte trasera ocupando el espacio libre en el sistema de captura del vaso trasero, estos han sido colocados lo más cerca posible de las ruedas. Este sistema de lastre que está formado por unas barras de acero, cumple también una segunda misión rigidizando más la base del robot. Ver Figura 3-4.

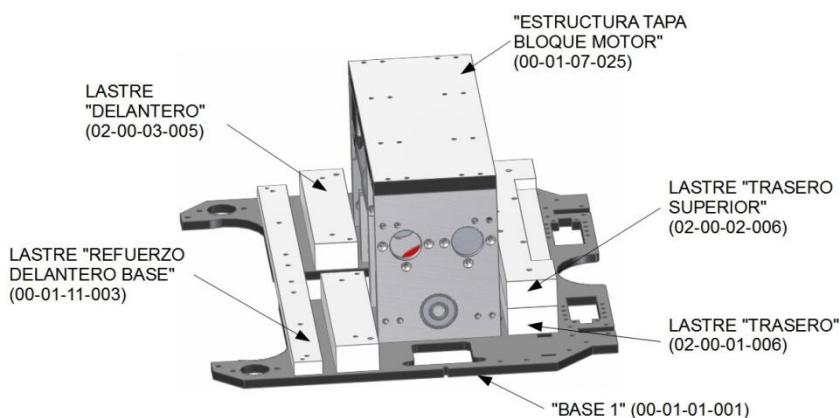


Figura 3-4 Disposición sistema de lastre en base robot en torno a bloque motor

### 3.3. Sistema de tracción

El sistema de tracción está compuesto por dos motores brushless Dunkermotoren BG40x50 con reductora y controladora incorporadas. La utilización de estos motores no ha sido una elección del diseño sino que ha sido impuesta como especificación de diseño como aparece también en el pliego de condiciones. Estos motores disponen de cuatro tornillos de sujeción tanto en la parte trasera como delantera que han sido utilizados para armar en parte el conjunto mostrado en la Figura 3-1. Digo solo en parte porque como se ha explicado en la sección de la estructura, es esta la que termina de armar el conjunto del bloque motor.

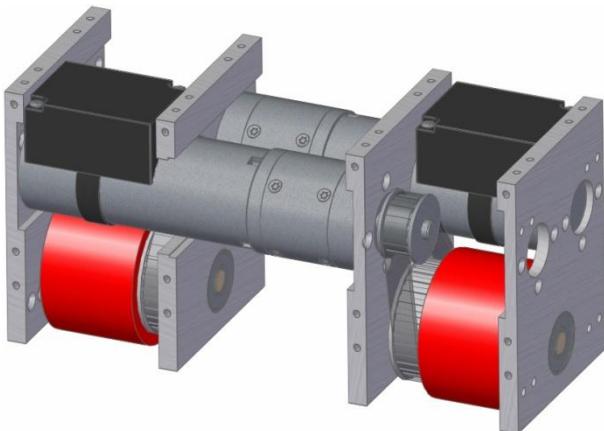


Figura 3-5 Vista perspectiva  
bloque motor

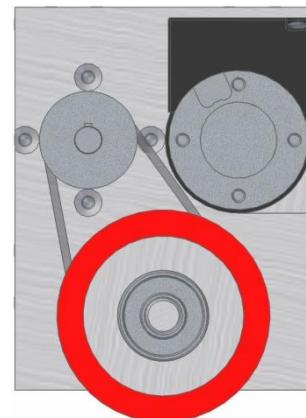


Figura 3-6 Vista lateral bloque motor

Esta estructura la completa una etapa de reducción creada con dos poleas dentadas y una correa por cada motor, que ha sido necesaria para adaptar las revoluciones del motor al rango total de velocidades aplicables para el desplazamiento en la prueba. A continuación voy a resumir los cálculos realizados para el diseño de esta etapa.

Especificaciones de partida para el diseño que han sido impuestas en el apartado 2.1 Especificaciones de diseño, en la secuencia de movimiento 1-3.

$$V_{max} = 1 \text{ m/s (velocidad máxima en línea recta)}$$

$$a_{max} = 3.6 \text{ m/s}^2$$

*Características motor Dunkermotoren BG40x50:*

$$w_{M\text{-max(continua)}} = 3640 \text{ rpm}$$

$$M_{W\text{-max}} = 8 \text{ Ncm (Para revoluciones máximas continuas)}$$

$$I_{max(\text{continua})} = 2.1 \text{ A (Corriente máxima continuo)}$$

$$M_{\text{arranque}} = 37.4 \text{ Ncm (Par máximo continuo)}$$

*Características reductora incorporada:*

$$i_1 = 4 (\text{Ratio de reducción})$$

$$M_{max(\text{continua})} = 350 \text{ Ncm (Par máximo continuo)}$$

$$\eta = 0.9 (\text{Eficiencia})$$

*Características finales de diseño empleadas en algunos cálculos:*

$$R_{rueda} = 29.3\text{mm} (\text{Ratio rueda con cubierta bajo compresión de peso final de robot})$$

$$P_{ROBOT} = 14.5\text{Kg} (\text{Peso final de robot})$$

### Ratio de reducción de segunda etapa

Como punto de partida he calculado el ratio de reducción de la segunda etapa para ajustar el rango completo de revoluciones del motor al rango de velocidad de movimiento lineal del robot.

Aunque la velocidad máxima estimada para la trayectoria 1-3 se ha fijado en 1 m/s, para no trabajar con el motor a máxima potencia de manera permanente la reductora va a ser dimensionada para permitir una velocidad máxima de 1.4 m/s.

$$\omega_{M-\max} = 3640(\text{rev/min}) * (2\pi/60) = 381.17(\text{rad/s})$$

$$\omega_{Rueda-\max} = \frac{V_{\max}}{R_{Rueda}} = \frac{1.4(\text{m/s})}{0.0293(\text{m})} = 47.78(\text{rad/s})$$

$$i_T = i_{\text{Etapa 1}} * i_{\text{Etapa 2}} = \frac{\omega_{M-\max}}{\omega_{Rueda-\max}}$$

$$i_{\text{Etapa 2}} = \frac{\omega_{M-\max}}{\omega_{Rueda-\max} * i_{\text{Etapa 1}}} = \frac{381.17}{47.78 * 4} = 1.9$$

El ratio de reducción de la segunda etapa finalmente se va a redondear a 2. Por cada dos vueltas del eje de la primera reductora solo se va a transmitir 1 al eje de las ruedas. Esto va a permitir que los motores trabajando a máxima potencia puedan desplazar el robot a 1.4 m/s.

### Elección de sistema de transmisión

El sistema elegido ha sido una transmisión por correa dentada. A continuación se detallan las principales razones que han decidido este sistema frente a la transmisión por engranajes.

- La distancia entre ejes en torno a 55 mm permite una transmisión por poleas dentadas con un módulo suficiente para la potencia a transmitir. Una distancia demasiado pequeña limitaría el módulo máximo respetando el número mínimo de dientes que deben ser engranados por polea. Este número no debe ser menor de 6 dientes, aunque este dato varía según el fabricante de la polea.
- Esta distancia de transmisión entre ejes encarecería de forma muy importante su realización con engranajes, frente al coste con poleas.
- Las poleas son un sistema mucho más silencio, que además no requiere de lubricación. En general se ha evitado utilizar cualquier componente que precise de lubricación.

- El ajuste entre centros es menos crítico y además este tipo de poleas síncronas no requieren de elementos tensores.
- El rendimiento en términos generales alcanza valores mayores en los sistemas de poleas dentadas. La correa Synchrobelts de poliuretano utilizada tiene un rendimiento del 98%. Lo que va a suponer una pérdida de potencia mínima en la transmisión, sin tener en cuenta las pérdidas en los rodamientos.

Para la elección de la polea se ha calculado la potencia máxima en el eje primario de entrada en la segunda etapa. La potencia mecánica máxima que puede entregar el motor se produce cuando el par se encuentra en la mitad de su rango máximo de variación. En la Figura 3-9 se muestra las curvas del motor en función del par motor a tensión nominal constante.

$$M_{\text{POTENCIA MAX}} = \frac{M_{W-\text{max}}}{2} = \frac{40(\text{Ncm})}{2} = 20(\text{Ncm}) w_{M_{\text{POTENCIA MAX}}} = 2250(\text{rpm})$$

$$P_{\text{MAX}} = M_{\text{POTENCIA MAX}} * w_{M_{\text{POTENCIA MAX}}}$$

$$P_{\text{MAX}} = 20(\text{Ncm}) * 10^{-2}(\text{m/cm}) * 2250(\text{rev/min}) * 2\pi(\text{rad/rev}) * 60^{-1}(\text{min/s}) = 47.12\text{W}$$

$$P_{\text{eje\_primario}} = P_{\text{MAX}} * \eta_{\text{Etapa-1}} = 47.12\text{W} * 0.9 = 42.4\text{W}$$

El fabricante de la correa utilizada dispone de la siguiente gráfica para la elección de la correa adecuada en función de las revoluciones de funcionamiento y la potencia transmitida. Como se puede ver la gráfica comienza su escala para potencias de mínimo 100W, por lo tanto para nuestro caso no queda claramente definido y debido al amplio rango de velocidades de funcionamiento se ha decidido el uso del paso de 5 mm entre dientes. Se trata de una polea de paso métrico tipo "T".

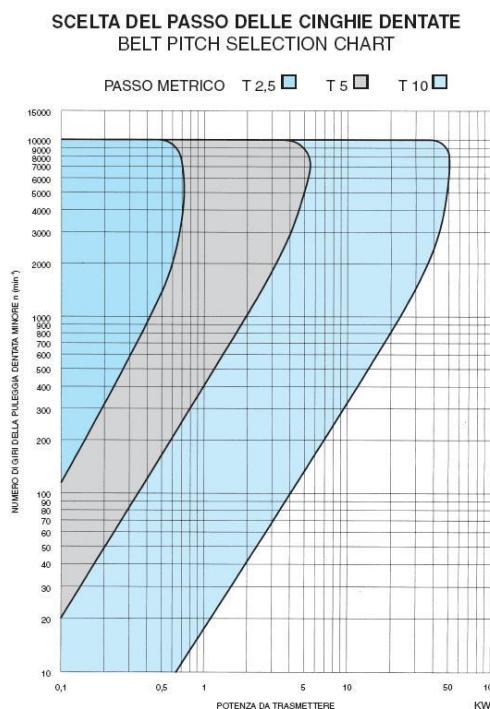


Figura 3-7 Grafica selección correas dentadas paso métrico serie "T" con curva Potencia-RPM (BEA TRANSMISION)

Con la relación de transmisión calculada en el anterior punto se han elegido dos poleas de 15 y 30 dientes:

$$i_{\text{Etapa2}} = \frac{Z_2}{Z_1} = \frac{30}{15} = 2$$

El fabricante de estas poleas “BEA Transmisión” dentro de las especificaciones no indica el diámetro primitivo, por lo que ha sido necesario calcularlo:

$$d_{P1} = \frac{\text{PASO} * \text{DIENTES}}{\pi} = \frac{5 * 15}{\pi} = 23.87\text{mm (Diámetro primitivo Polea Entrada)}$$

$$d_{P2} = \frac{5 * 30}{\pi} = 47.74\text{mm (Diámetro primitivo Polea Salida)}$$

### Desarrollo de correa estándar y cálculo de distancia entre ejes

Los fabricantes de poleas como es lógico solo ofrecen ciertas longitudes estandarizadas. Por lo tanto es necesario una vez concluida la primera aproximación a la distancia entre centros según las necesidades del diseño, calcular el desarrollo obtenido y seleccionar el estándar más próximo. Y posteriormente con este desarrollo estándar recalcular la distancia entre centros correcta.

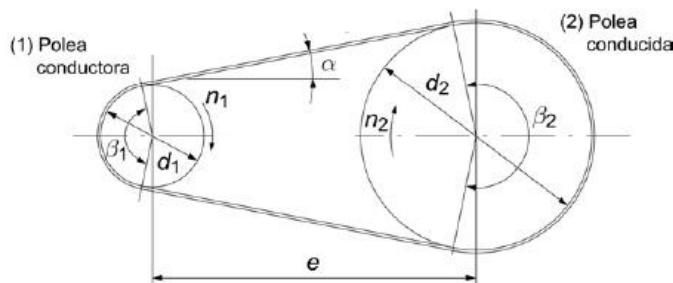


Figura 3-8 Calculo desarrollo polea (Zayas Figueras, 2008)

$$\text{Longitud}(L) = 2 * e * \cos\alpha + \frac{(d_1 + d_2) * \pi}{2} + (d_2 - d_1) * \alpha(\text{rad})$$

Calculo del desarrollo de la correa con una distancia inicial de diseño entre ejes de 60 mm:

$$\operatorname{sen}\alpha = \frac{(d_2 - d_1) * \pi}{2 * e} = \frac{(47.74 - 23.87) * \pi}{2 * 60} \alpha = 38,67^\circ(0.675\text{rad})$$

$$L_{60} = 2 * 60 * \cos 38.67 + \frac{(23.87 + 47.74) * \pi}{2} + (47.74 - 23.87) * 0.675 = 227.12\text{mm}$$

La longitud estándar más próxima es de 225 mm. Por lo tanto se calcula la longitud entre centros para este desarrollo:

$$H = \frac{L}{4} - 0.3925(d_2 + d_1) = \frac{225}{4} - 0.3925(47.74 + 23.87) = 28.143$$

$$B = \frac{(d_2 - d_1)^2}{8} = \frac{(47.74 - 23.87)^2}{8} = 71.22$$

$$e = H + \sqrt{(H^2 - B)} = 28.143 + \sqrt{(28.143^2 - 71.22)} = 54.99\text{mm}$$

Finalmente la distancia es ajustada a 54.99 mm.

### Estudio de aceleración máxima

Dentro de las especificaciones del diseño se han establecido junto con la velocidad máxima necesaria también la aceleración y deceleración máxima objetivo. El perfil de velocidad que se va a emplear en los movimientos del robot siempre va a tener forma trapezoidal, pudiéndose descomponer en tres fases:

- Fase de aceleración, con aceleración constante (movimiento uniformemente acelerado).
- Fase de avance a velocidad constante, igual o menor a velocidad máxima de la especificación.
- Fase de frenado, con deceleración constante (movimiento uniformemente decelerado).

Las aceleraciones óptimas serán aquellas que estén al límite del patinaje de las ruedas despreciando el efecto caballito por el propio diseño del robot que combina un eje motriz centrado con apoyos delanteros y traseros suficientemente distanciados para eliminar este efecto, por lo tanto siempre será más bajo el límite por patinaje de las ruedas.



Figura 3-9 Curva motor a alimentación nominal (Hubrich, 2006)

Ecuación de la curva de carga del motor (PAR(T)-REVOLUCIONES(N)):

$$\frac{T - T_1}{T_2 - T_1} = \frac{N - N_1}{N_2 - N_1}$$

$$\frac{T - 37.5}{8 - 37.5} = \frac{N - 0}{3640 - 0}$$

$$T = 37.5 - \frac{29.5 * N}{3640}$$

Calculo de potencia en el eje de la rueda:

$$P_{RUEDA} = \eta_{\text{Etapa 1}} * \eta_{\text{Etapa 2}} * P_{MOTOR} = 0.9 * 0.9 * P_{MOTOR} = 0.81 * M_{MOTOR} * w_{MOTOR}$$

Calculo de velocidad en el eje de la rueda:

$$i_T = i_{\text{Etapa 1}} * i_{\text{Etapa 2}} = \frac{w_M}{w_{\text{Rueda}}} w_{\text{Rueda}} = \frac{w_M}{i_T} = \frac{w_M}{8}$$

Calculo de par en el eje de la rueda:

$$M_{RUEDA} = \frac{P_{RUEDA}}{w_{RUEDA}} = \frac{0.81 * M_{MOTOR} * w_{MOTOR}}{\frac{w_{MOTOR}}{8}} = 6.48 * M_{MOTOR}$$

$$M_{MOTOR} (\text{Ncm}) = 37.5 - \frac{29.5 * w_{MOTOR} (\text{rpm})}{3640}$$

Sustituyendo:

$$M_{RUEDA} = 6.48 * (37.5 - \frac{29.5 * w_{MOTOR} (\text{rpm})}{3640})$$

$$M_{RUEDA} = 6.48 * (37.5 - \frac{29.5 * 8 * w_{RUEDA} (\text{rpm})}{3640})$$

Calculo de aceleración tangencial máxima de la rueda (aceleración del robot) en función de la velocidad de avance del robot:

$$F_{\text{TANGENCIAL\_RUEDA}} = \frac{M_{RUEDA}}{R_{\text{RUEDA}}}$$

$$F_{\text{TANGENCIAL\_TOTAL}} = 2 * F_{\text{TANGENCIAL\_RUEDA}} (\text{Fuerza total sistema 2 motores - 2 ruedas})$$

$$a_{ROBOT} = \frac{F_{\text{TANGENCIAL\_TOTAL}}}{Masa_{ROBOT}} = \frac{2 * (\frac{M_{RUEDA}}{R_{\text{RUEDA}}})}{Masa_{ROBOT}}$$

$$a_{ROBOT} = \frac{2 * \left( \frac{6.48 * \left( 37.5 - \frac{29.5 * 8 * w_{RUEDA} (\text{rpm})}{3640} \right)}{R_{\text{RUEDA}}} \right)}{Masa_{ROBOT}} =$$

$$\frac{2 * 6.48}{R_{\text{RUEDA}} * Masa_{ROBOT}} * \left( 37.5 - \frac{29.5 * 8 * w_{RUEDA} (\text{rpm})}{3640} \right)$$

$$\text{Siendo: } w_{RUEDA} (\text{rpm}) = \left( \frac{v_{\text{TANGENCIAL\_RUEDA}} (\text{m/s})}{R_{RUEDA} (\text{m})} \right) (\text{rad/s}) * 60(\text{s/min}) * (2\pi)^{-1} (\text{rev/rad})$$

$$a_{ROBOT} (\text{m/s}^2) = \frac{2 * 6.48}{2.93 * 14.5(\text{kg})} * (37.5 - \left( \frac{29.5 * 8 * 60 * v_{\text{TANGENCIAL\_RUEDA}} (\text{m/s})}{3640 * 0.0293 * 2 * \pi} \right))$$

$$\text{Siendo: } v_{\text{ROBOT}} (\text{m/s}) = v_{\text{TANGENCIAL\_RUEDA}} (\text{m/s})$$

$$a_{ROBOT} (\text{m/s}^2) = 11.4393 - 6.4458 * v_{\text{TANGENCIAL\_RUEDA}} (\text{m/s})$$

A partir de esta ecuación se puede comprobar que los motores serán capaces de producir una aceleración superior a los  $3.6 \text{ m/s}^2$  de la especificación hasta que el robot alcance una velocidad lineal de  $1.21 \text{ m/s}$  (ver siguiente cálculo), cumpliéndose la mínima requerida en la especificación que es de  $1 \text{ m/s}$ .

$$v_{\text{ROBOT}} (\text{m/s}) = \frac{11.4393 - a_{\text{ROBOT}}}{6.4458} = \frac{11.4393 - 3.6(\text{m/s}^2)}{6.4458} = 1.21(\text{m/s})$$

Por lo tanto se puede concluir que los motores y la etapa de reducción no solo cumplen los requerimientos cinemáticos mínimos de la especificación, sino que además están correctamente dimensionados para ella, teniendo en cuenta que han sido una imposición para el diseño.

Calculo de velocidad en función del tiempo para la máxima aceleración:

$$\text{Siendo: } dv = adt$$

$$\begin{aligned} dv &= (11.43 - 6.44v)dt \\ dv &= 6.44(1.77 - v)dt \\ \frac{dv}{(1.77 - v)} &= 6.44dt \end{aligned}$$

Integrando a ambos lados:

$$\begin{aligned} \int_{v_0}^v \frac{dv}{(1.77 - v)} &= \int_{t_0}^t 6.44dt \\ -\ln|1.77 - v| - (-\ln|1.77 - v_0|) &= 6.44t - 6.44t_0 \end{aligned}$$

En caso de  $t_0 = 0$  y  $v_0 = 0$ :

$$v = 1.77(1 - e^{-6.44t}) \quad ; \quad t = \frac{\ln(\frac{1.77}{1.77 - v})}{6.44}$$

A través de estas ecuaciones es posible calcular el tiempo necesario para alcanzar la velocidad máxima de la especificación, si no se limitara por software con las rampas de

aceleración creadas, la máxima alcanzable. Siendo este de 0.129 segundos (ver siguiente cálculo)

$$\text{Para } v = 1(m/s) \rightarrow t = \frac{\ln(\frac{1.77}{1.77-1})}{6.44} = \frac{\ln(\frac{1.77}{1.77-1})}{6.44} = \frac{\ln(2.298)}{6.44} = 0.129 \text{ seg}$$

Evidentemente en la práctica no es posible alcanzar este tiempo, ya que como se deduce de la ecuación  $\{a_{ROBOT}(m/s^2) = 11.4393 - 6.4458 * v_{ROBOT}(m/s)\}$ , la aceleración en el instante inicial alcanzaría los  $11.43 \text{ m/s}^2$ , y como se va a calcular en el apartado 7.1 "Estimación de masa y posición de centro de masas" dentro del cálculo de la aceleración límite para el patinaje de las ruedas, la aceleración está limitada a  $9.94 \text{ m/s}^2$  en la dirección de avance y  $7.57 \text{ m/s}^2$  en la de retroceso.

Calculo de espacio recorrido en función del tiempo para la máxima aceleración:

$$\text{Siendo: } ds = vdt$$

Integrando a ambos lados:

$$s = \int_{t_0}^t 1.77(1 - e^{-6.44t}) = (1.77t + \frac{1.77}{6.44}e^{-6.44t}) - (1.77t_0 + \frac{1.77}{6.44}e^{-6.44t_0})$$

*En caso de  $t_0 = 0$ :*

$$s = 1.77t + \frac{1.77}{6.44}e^{-6.44t} - \frac{1.77}{6.44}$$

Con esta última ecuación es posible calcular el espacio recorrido durante el tiempo de aceleración hasta la velocidad de la especificación manteniendo la aceleración máxima.

### 3.4. Sistema de posicionamiento odométrico

---

Las especificaciones previas definidas para el diseño de este sistema han sido las siguientes:

- Se deben utilizar los Encoder RI41 Hengstler 3600 pulsos.
- Deben ser colocados en el mismo eje que las ruedas de tracción.
- Las ruedas del encoder se colocaran tan próximas como sea posible a los laterales del robot para mejorar la precisión del sistema odométrico.

Este sistema va a estar constituido por dos encoder independientes de los motores de tracción que van a permitir la implementación de un sistema de control de posición por odometría, creando un sistema de control en lazo cerrado.

Los robots móviles usan la odometría para estimar su posición relativa a su localización inicial. Es bien sabido que la odometría proporciona una buena precisión a corto plazo. Sin embargo la idea fundamental de la odometría es la integración de información incremental del movimiento a lo largo del tiempo, lo cual conlleva una inevitable acumulación de errores. En concreto, la acumulación de errores de orientación, causa grandes errores en la estimación de la posición, los cuales van aumentando proporcionalmente con la distancia recorrida por el robot. (Wikipedia)

Para minimizar en lo posible los errores de posicionamiento acumulativo se han tenido en cuenta las siguientes consideraciones de diseño:

- Se ha independizado la realimentación de la posición de las ruedas de tracción colocando los encoder en ruedas independientes. De esta forma se evitan los errores por patinaje y por la propia morfología de la rueda (el ancho de la rueda).
- Se han utilizado unas ruedas de encoder con una cubierta de goma en forma de toroide. De esta forma el punto de contacto de la rueda con el suelo y su distancia al centro del robot es más precisa y sobre todo constante.
- Se ha diseñado un sistema de apoyo con fuerza constante que asegura el contacto de las ruedas de los encoder en todo momento con el piso con independencia de las irregularidades.
- Todas las piezas fabricadas para los bloques intervenientes en el sistema de tracción y odometría han sido fabricadas por mecanizado por control numérico minimizando los errores de alineamiento entre las cuatro ruedas del sistema. Así mismo el sistema de montaje entre piezas perpendiculares mediante pestañas insertables ha sumado para el aumento de esta garantía.

En general la mayoría de los errores de construcción como pueden ser la variación de la distancia entre las ruedas, la variación del diámetro de las diferentes ruedas y todos los demás que ocasionan errores fijos o sistemáticos, pueden ser corregidos por procesos de calibración software, en la fase de ajuste del robot. Por el contrario los errores no sistemáticos representan el mayor inconveniente y punto crucial a evitar en el diseño. Más adelante voy a hacer incisos sobre este aspecto en algunos detalles del diseño utilizado.

Este sistema no pasaría de ser un simple acoplamiento para un encoder y su rueda si no fuera por la particularidad de la necesidad de garantizar el apoyo constante con el suelo. Para conseguir esto se ha utilizado un sistema oscilante articulando el movimiento mediante un brazo que pivota sobre un eje. En la Figura 3-10 se muestra el conjunto del brazo con la representación del encoder con la junta tórica y el eje de articulación del movimiento pivotante.

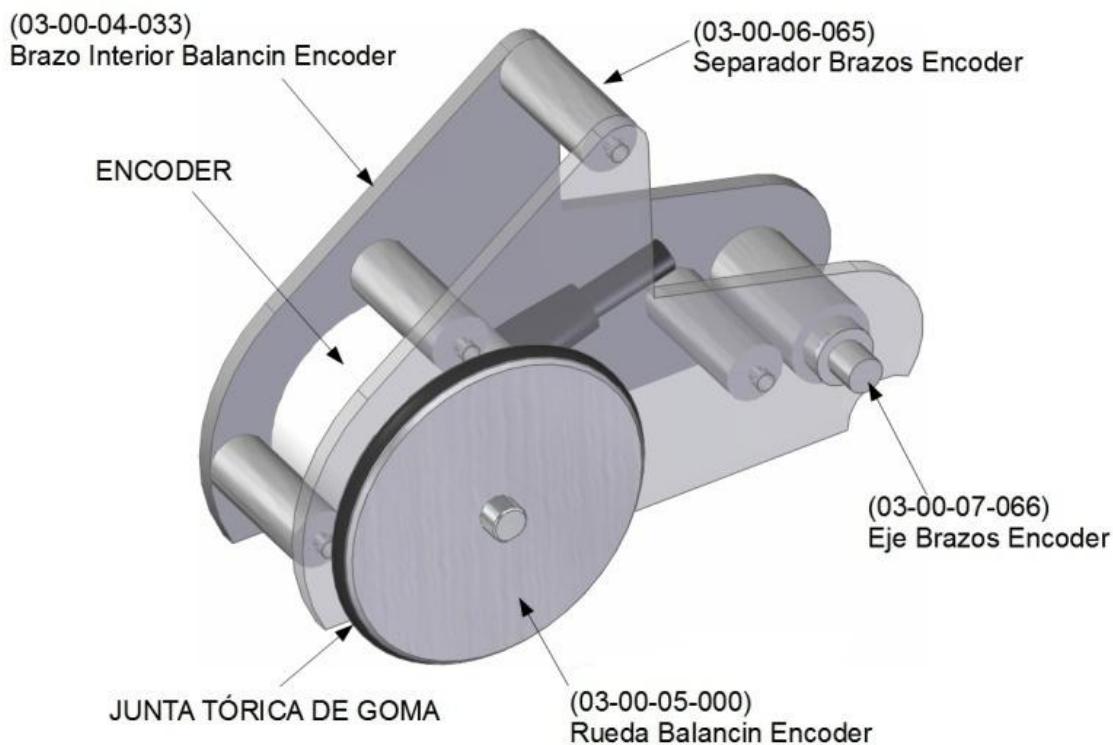


Figura 3-10 *Conjunto balancín*

La elección de este sistema radica en la simplicidad de la construcción. Aunque pese a esto presenta una principal desventaja que ha sido necesario estudiar para su definitiva aceptación. El movimiento giratorio sobre un eje en el mismo plano horizontal (mejor de los casos) produce desalineamientos del eje del encoder con el eje de la rueda de tracción, que van a producir errores no sistemáticos no deseados. Para minimizar el error es necesario aumentar la distancia entre los ejes lo máximo posible. Se ha calculado mediante una resolución gráfica con el mayor distanciamiento realizable entre ejes 65 mm, la desviación ocasionada ante un supuesto caso de elevación o descenso del eje del encoder sobre el eje de la rueda de (27.5-26.37) 1.37 mm que corresponde a 1º de giro. Siendo esta de (65-69.9901) 0.0099 mm se ha considerado despreciable para la resolución de trabajo. Ver la Figura 3-11 y la Figura 3-12.

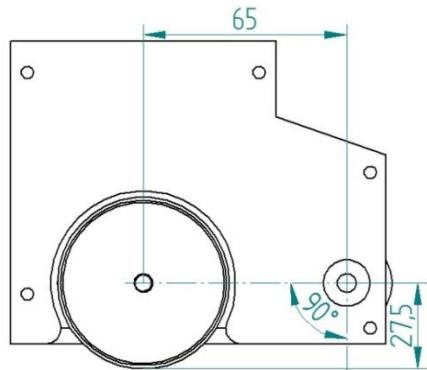


Figura 3-11 Brazo horizontal 90°

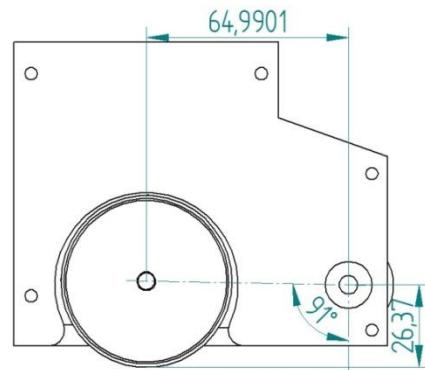


Figura 3-12 Brazo giro 1°

Incluyendo las consideraciones derivadas del estudio anterior se enumeran todas las tenidas en cuenta para minimizar los errores no sistemáticos.

- Máxima distancia entre eje de rueda de encoder y eje de giro de brazo.
- Eje de encoder y eje de brazo alineados en el mismo plano horizontal.
- Evitar cualquier posible holgura en la articulación de giro del brazo que pueda producir movimientos en el eje axial del encoder. Para evitar esto se ha diseñado un sistema de articulación robusto pivotante entre dos rodamientos con el mayor distanciamiento posible en el espacio de diseño.

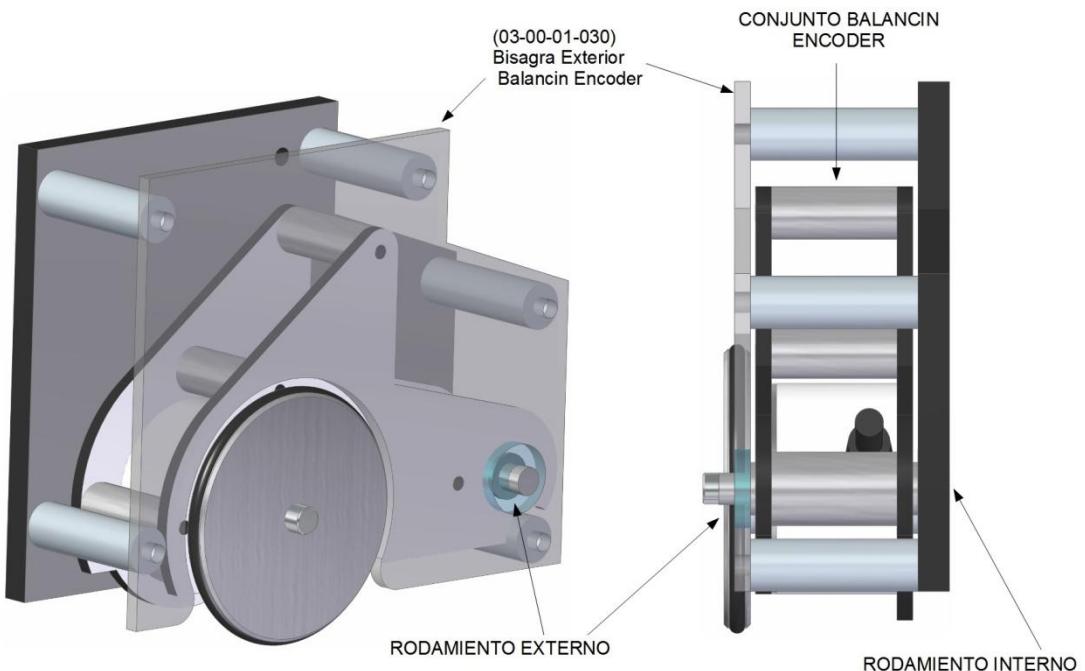


Figura 3-13 Perspectiva Conjunto Bisagra y Balancín Encoder, a la izquierda perspectiva y a la derecha vista lateral

En la Figura 3-14 se muestra la ubicación del conjunto del bloque del encoder en el lateral del robot. En esta imagen se ha retirado la tapa lateral derecha.

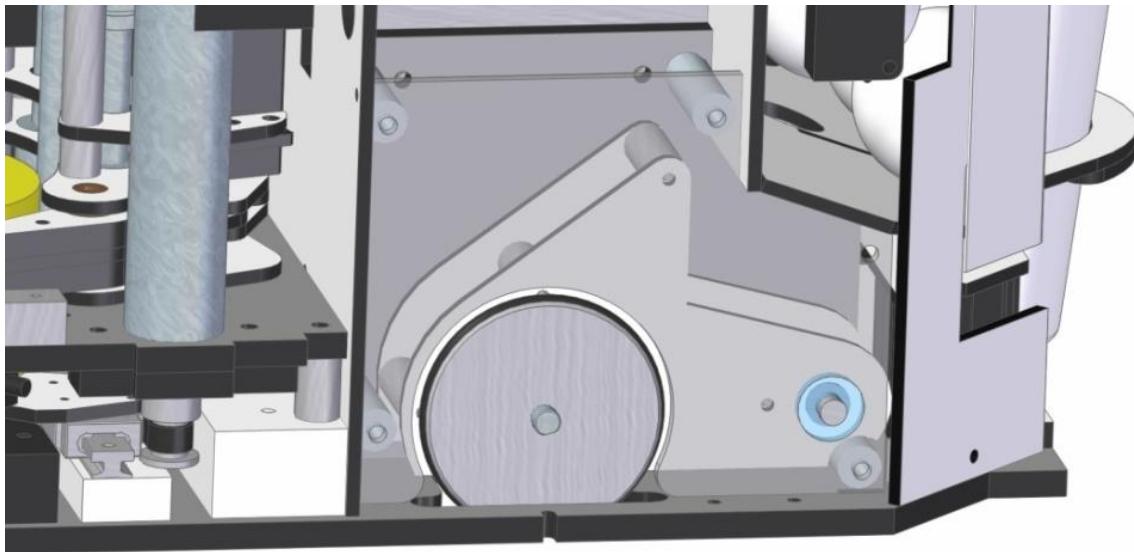


Figura 3-14 Sistema encoder en robot vista lateral

En la Figura 3-14 se muestra la abertura realizada en la base del robot para la rueda del encoder. Debajo de la rueda del encoder en color blanco se ve el encoder y más abajo en rojo la rueda de tracción e inmediatamente más abajo la polea y correa de tracción.

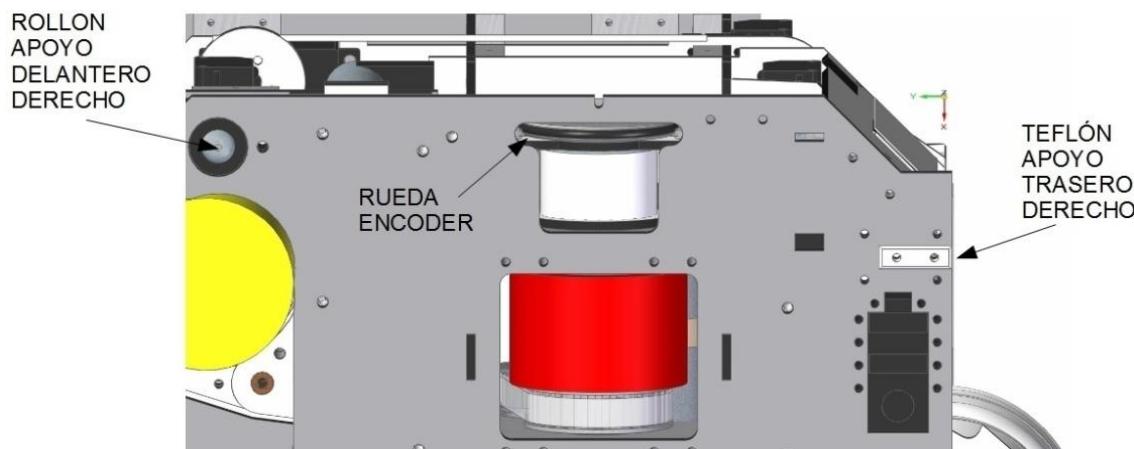


Figura 3-15 Abertura encoder en base robot

## 4. Sistema de manipulación de stands

---

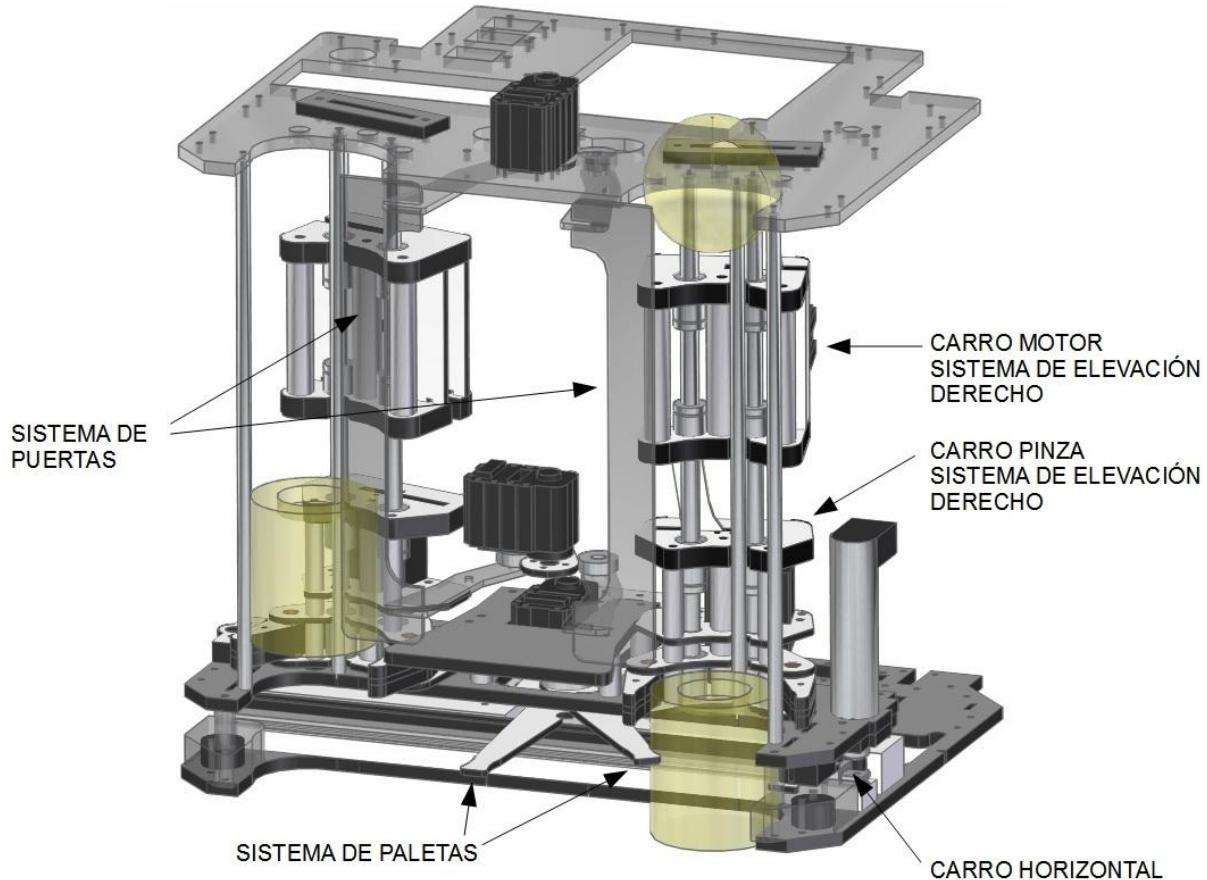


Figura 4-1 *Sistemas de manipulación de stands*

En este capítulo se recoge el diseño realizado para el sistema de manipulación de stand. Este sistema como se va a detallar en los próximos apartados está compuesto por cuatro subsistemas que deben ser controlados de manera conjunta para realizar todas las operaciones de manipulación de stand definidas en las especificaciones previas. Los cuatro subsistemas son los siguientes:

- **SISTEMA DE PALETAS DE STAND**

Este sistema va a permitir bloquear los stands que se introduzcan en la boca del robot durante su movimiento de avance, inmovilizándolos en una posición exacta. Este sistema es doble, con dos paletas con accionamiento independiente.

- **SISTEMA DE ELEVACIÓN DE STAND**

El almacén vertical de stands una vez inmovilizados por el sistema de paletas los bloquea mediante un sistema de pinzas y los eleva para permitir el bloqueo de un nuevo stand por el sistema de paletas.

La altura de apilamiento máximo es físicamente ilimitada y solo está limitada por la normativa de juego y la potencia del sistema de accionamiento. Este sistema es doble e independiente.

- **SISTEMA CARRO HORIZONTAL DE STAND**

Este sistema permite desplazar el stand desde una posición de bloqueo de paleta a la contraria o viceversa. Permite intercambiar los stands entre los almacenes verticales para la construcción del foco de 8 niveles de apilamiento.

- **SISTEMA DE PUERTAS DE STAND**

Este sistema tiene una doble función. Por un lado es uno de los tres puntos de guiado de los stands en el almacén vertical, los otros dos son fijos y por otro mediante su articulación permite la liberación de los stands apilados. Este sistema es doble habiéndose implementado uno para cada almacén.

Dependiendo de la fase de juego los sistemas utilizados son los siguientes:

- **RECOLECCIÓN DE STANDS EN ALMACÉN DERECHO O IZQUIERDO**  
Solo son utilizados el sistema de paletas y el sistema de elevación. El sistema de puertas solo interviene como barrera sin ser accionado.
- **CONSTRUCCIÓN DE FOCO**  
En este proceso intervienen el sistema de paletas, el intercambiador horizontal y ambos sistemas de elevación.
- **LIBERACIÓN DE FOCO O TORRE INDEPENDIENTE**  
Intervienen tanto el sistema de puertas como el de paletas, que siempre es el encargado de sujetar el stand a nivel del suelo.

Las especificaciones de este sistema que se detallan dentro de diferentes secuencias de la estrategia inicial del apartado 2.1, las he agrupado en los siguientes puntos:

- El sistema debe ser capaz de almacenar los stands que se introduzcan en el movimiento de avance sin detener dicho movimiento.
- Como ya se ha desarrollado y dado solución en el apartado 3.2 "Estructura principal y sistema de ensamblado por pestañas" debe ser capaz de recoger los dos stands que se encuentran a los lados de la trayectoria 1-3.
- Se crearan dos sistemas de almacenamiento simétricos con funcionamiento independiente.
- Cada almacenen de recolección de manera independiente tendrá que tener un tiempo máximo de procesado de stand no superior al tiempo de desplazamiento entre las posiciones 2 y 4 (Ver Figura 2-1 *Secuencia de movimientos de estrategia optimizada sobre imagen* ). Este tiempo debe calcularse según las siguientes velocidades y aceleraciones:
  - Velocidad 1 m/s
  - Aceleración 3.6 m/s<sup>2</sup>
  - Velocidad angular 300 grados/seg
  - Aceleración angular 1500 grados/seg<sup>2</sup>
- También se tendrá que tener en cuenta el tiempo de la maniobra de captura del vaso que se ha estimado en 0,2 segundos.
- El tiempo de maniobra de bloqueo de stand no debería superar los 0,3 segundos, que corresponde con el tiempo de giro de un servo Dynamixel AX12 para 90 grados.
- El sistema será capaz de construir una torre apilando 8 stands con una pelota de tenis en la parte superior, que llevara precargada antes del inicio del partido.

## 4.1. Prototipos y estudio previo

---

De los cuatro subsistemas diseñados tan solo el sistema de elevación y almacenaje de stands ha necesitado un proceso de estudio más laborioso debido a su complejidad.

Dentro del sistema de elevación que está compuesto a su vez por dos subsistemas, un sistema de agarre de los stands y otro propiamente de elevación del stand agarrado, ha sido este último el que ha presentado la mayor variedad de alternativas para su resolución.

Las alternativas bajo estudio, con sus principales características según el sistema de accionamiento han sido las siguientes:

### 1. CILINDRO NEUMÁTICO.

Ventajas:

- Muy alta velocidad de movimiento.
- Alta capacidad de carga.
- Auto frenado de carga sin consumo de energía.

Inconvenientes:

- Necesidad de espacio para depósito de aire comprimido y válvulas de pilotaje.
- Limitación de presión máxima de depósito según normativa a 4 bares.
- Dificultad para controlar la velocidad de aceleración y deceleración del sistema.

### 2. MOTOR DC con sistema de husillo y tuerca.

Ventajas:

- Auto frenado de carga.
- Alta capacidad de carga.
- Posibilidad de control de aceleraciones.

Inconvenientes:

- Necesidad de driver de control y complejidad añadida.
- Ratio de reducción elevado con requerimiento de motor de alta velocidad.
- Necesidad de transmisión entre motor y husillo.
- Bajo rendimiento.

### 3. MOTOR DC con sistema de correa.

Ventajas:

- Alta velocidad de movimiento.
- Posibilidad de control de aceleraciones.

Inconvenientes:

- Necesidad de driver de control y complejidad añadida.
- Motor bajo carga constante

### 4. SERVOMOTOR AX12 con sistema de palanca y leva.

Ventajas:

- Sencillez de implementación.
- Sencillez de control.
- Auto frenado de carga según diseño.
- Control de aceleraciones de forma mecánica según diseño.
- Bajo coste.

Inconvenientes:

- Limitación de par máximo para accionamiento.

El proceso de elección de este sistema que ido evolucionando a medida que se iban realizando pruebas y analizando cada uno de las ventajas e inconvenientes de cada una de las soluciones enumeradas, comenzando desde la 1 hasta concluir con la elección y el diseño definitivo de la 4.

Según las especificaciones de velocidad detalladas que van a ser estudiadas más adelante la principal característica que primó inicialmente fue la velocidad del sistema elegido. Se optó inicialmente por un sistema neumático, aun conociendo la complejidad añadida de restar al robot el espacio necesario para el alojamiento del depósito de aire comprimido y las válvulas de pilotaje. Este espacio se estimó en  $1000 \text{ cm}^3$  en función del número máximo de maniobras por partida que es de 20 y el tamaño del cilindro elegido que fue de 10 mm de diámetro interior por una carrera de 75 mm. El cálculo del diámetro del embolo del cilindro se realizó en función de la carga máxima de elevación que es de 7 stands (el octavo nunca se eleva) más el peso de una pelota de tenis estándar. A continuación se muestra el cálculo del peso total realizado, suponiendo que se iba a utilizar una madera con una densidad no superior a la del pino (este dato no está detallado en la normativa y en ningún caso se esperaba que se utilizara una madera más noble y por lo tanto con mayor densidad).

$$Masa_{stand} = d \times v = 0.5(\text{kg}/\text{dm}^3) \times \pi \times 0.3^2 \times 0.7 = 0.098\text{Kg} \approx 0.1\text{Kg}$$

$$Masa_{elevación_{MAX}} = (7 \text{ stand} \times 0.1(\text{kg})) + 0.058(\text{kg}) = 0.758 \text{ Kg}$$

$$F_{Resistente\_elevacion\_MAX} = Masa_{elevación_{MAX}} \times g = 7.428 \text{ N}$$

Una vez conocido el peso máximo a elevar se estimó la fuerza mínima necesaria en función de la caída de presión tras 20 ciclos de movimiento que también es función del diámetro del embolo elegido.

Definido el diámetro del cilindro se concluyó con la estimación del tamaño del depósito necesario, para que la caída de presión máxima no comprometiese el último movimiento. Recurrentemente he hablado de estimación ya que no se tuvieron en cuenta las pedidas por el volumen de aire en las conducciones y las válvulas de pilotaje, así como por las conmutaciones de dichas válvulas que como se demostró en las pruebas del prototipo construido para la simulación de este sistema eran muy importantes.

Como he adelantado se realizó un prototipo funcional para la prueba del sistema. No se incluyen fotos de este sistema pero si se ha adjuntado un video de una simulación del funcionamiento y la caída de presión resultante tras 20 ciclos. Este video se encuentra en la ruta del soporte digital *VIDEOS\PROTOTIPOS\ELEVADOR NEUMATICO\Sistema elevador neumático.mp4*.

Aunque las conclusiones obtenidas evidenciaron que la caída de presión era muy superior a la esperada, este no fue el aspecto fundamental para desestimar este sistema, fue la dificultad para controlar la desaceleración del movimiento de ascenso, necesario para evitar que los stands durante la fase de construcción y una vez sobrepasada la altura de guiado de la torre por encima de la cota máxima del robot se desalinearan con posibilidad de caída de la torre. Además tras las pruebas se evidencio la necesidad de disponer de una posición intermedia de parada del desplazamiento de elevación entre la cota superior e inferior para

que durante la fase de construcción de la torre fuera posible optimizar los tiempos para liberar el stand inferior con un pequeño desplazamiento de elevación.

Tras esta experiencia se prestó especial importancia durante la fase de diseño al control de la aceleración y deceleración del movimiento de elevación.

Descartado el sistema neumático se continuó el diseño con sistemas accionados por motores de corriente continua inicialmente por transmisión de husillo y posteriormente con un sistema de transmisión por correa para simplificar la construcción, ambos sistemas por supuesto ya permitían un control de la aceleración mediante la generación de rampas de aceleración y deceleración.

Comenzando el diseño 3D de este último sistema se sopesó el espacio que ocupaba y la complejidad añadida para el posterior control de los motores con la implementación de dos controles PID. Además se añadía la necesidad de la ampliación de la electrónica de control que iba a ser utilizada y que solo estaba capacitada para controlar desde el micro esclavo dos motores DC, siendo el total necesario de tres ya que en este punto se conocía la necesidad de un tercer motor para el control del carro horizontal que necesariamente debía ser controlado por un sistema con las mismas características. En la Figura 4-2 se muestra el diseño que se implementó para esta última solución.

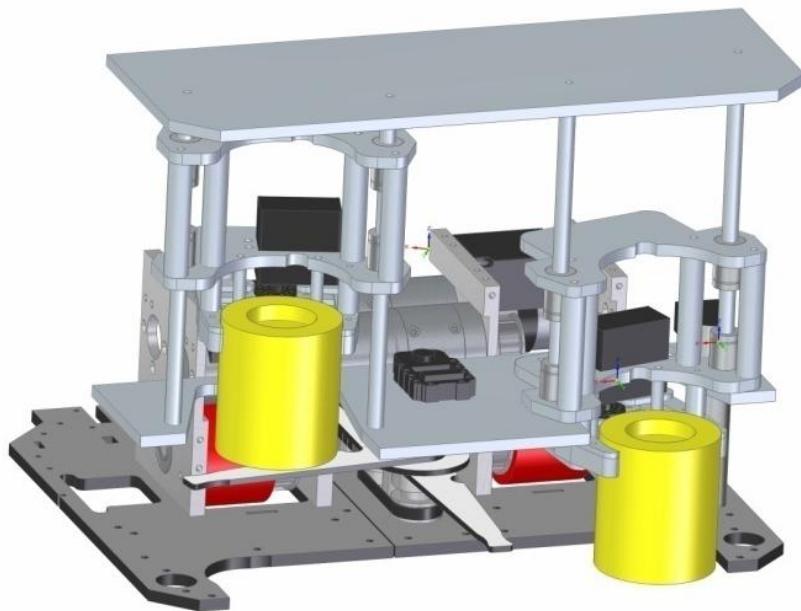


Figura 4-2 Diseño de alternativa de elevación por correa

Alcanzada este fase del diseño en la que sucesivas realimentaciones de las conclusiones previas fueron conformando las características que debía presentar el diseño definitivo, se acometido un diseño que debía cumplir y cumplió las siguientes características que enumeran por orden de importancia.

## **1. Control de la aceleración y deceleración del movimiento**

El sistema diseñado mediante un sistema de dos manivelas contrapuestas y dos bielas transforma un movimiento angular a velocidad constante en un movimiento lineal con una función fácilmente resoluble por cálculo trigonométrico y que va integrar funciones de seno o coseno del ángulo de giro, lo que va a generar un movimiento con una aproximación a una onda sinusoidal. Esto se va a traducir en una aceleración y deceleración variable en el tiempo y que se va a mantener durante todo el desplazamiento. El estudio realizado para el diseño de este sistema se va a desarrollar en el apartado de elevación de stands.

## **2. Sistema de control del movimiento sencillo y con pocos recursos**

Este se ha conseguido como consecuencia directa de la resolución del punto número 1, y la utilización de un servo AX12 con control por línea común TTL Multi Drop que ha simplificado notablemente las fases posteriores de desarrollo.

## **3. Rapidez de maniobra**

Este punto que partía como la premisa fundamental en la fase inicial del diseño, no se ha visto comprometida por esta solución ya que como se va a comprobar este servo con una velocidad máxima de 60 rpm, va a permitir con el diseño implementado una elevación con un movimiento de 180° y un tiempo de desplazamiento que se acercara al medio segundo.

## **4. Espacio reducido de implementación**

El sistema desarrollado es finalmente con diferencia el más compacto de todos los estudiados, ya que se ha utilizado para la motorización del carro de la pinza exclusivamente el espacio visto en planta inmediatamente superior a este. El esto de sistemas necesitaban espacio en torno a este carro para albergar los sistemas de elevación, en este caso se ha colocado inmediatamente encima.

## **5. Auto frenado de carga en posiciones estables con mínimo consumo**

El sistema de dos manivelas contrapuestas y dos bielas ha sido diseñado para que en las cotas de desplazamiento vertical máximo y mínimo, todos los ejes de giro se encuentren alineados con el eje motor, por lo que el momento resultante sobre este es cero.

## **6. Bajo coste y sencillez de construcción**

La utilización de este servo motor sin ningún elemento de transmisión adicional ha reducido notablemente el coste final y sobre todo a simplificado su construcción.

El mayor inconveniente que ha presentado este sistema como ya se apuntaba ha sido asegurar que tenía potencia suficiente para asegurar la velocidad de maniobra requerida. Este punto será desarrollado en el apartado correspondiente.

Para concluir este apartado se van a presentar los cálculos realizados del mínimo tiempo de maniobra de procesado de un stand exigido en las especificaciones.

El tiempo total se ha calculado como la suma del tiempo de desplazamiento y operación entre las etapas 2 y 4. Se han realizado los siguientes cálculos previos teniendo en cuenta que

se trata de movimientos con rampas de aceleración y deceleración constantes hasta alcanzar las velocidades máximas de avance y giro.

$$\text{Tiempo acel. y dec. lineal: } t_a = t_{-a} = \frac{v_{max}}{a_{cte}} = \frac{1(m/s)}{3.6(m/s^2)} = 0.27 \text{ seg}$$

$$\text{Despla. en acel. lineal: } d_a = d_{-a} = 0.5 \times a_{cte} \times t_a^2 = 0.5 \times 3.6 \times 0.27^2 = 0.131 \text{ m}$$

$$\text{Tiempo acel. y dec. angular: } t_\alpha = t_{-\alpha} = \frac{\omega_{max}}{\alpha_{cte}} = \frac{300(deg/s)}{1500(deg/s^2)} = 0.2 \text{ seg}$$

$$\text{Despla. en acel. angular: } \theta_\alpha = \theta_{-\alpha} = 0.5 \times \alpha_{cte} \times t_\alpha^2 = 0.5 \times 1500^\circ \times 0.2^2 = 30^\circ$$

Calculo de tiempo de desplazamiento entre la posición 2 a la posición 3 (321 mm), en la posición 2 el movimiento ya ha alcanzado la velocidad máxima de desplazamiento:

$$t_{2 \rightarrow 3} = t_{v_{cte}}(d_t - d_a) + t_{-a} = \frac{(d_t - d_a)}{v_{max}} + t_{-a} = \frac{(0.321(m) - 0.131(m))}{1(m/s)} + 0.27 \text{ (s)} = 0.46 \text{ seg}$$

Calculo de tiempo de giro en la posición 3 (giro a derechas de 126°):

$$\begin{aligned} t_3 &= t_\alpha + t_{\omega_{cte}}(\theta_t - \theta_\alpha - \theta_{-\alpha}) + t_{-\alpha} = t_\alpha + \frac{(\theta_t - \theta_\alpha - \theta_{-\alpha})}{\omega_{max}} + t_{-\alpha} \\ &= 0.2(s) + \frac{(126(deg) - 30(deg) - 30(deg))}{300(deg/s)} + 0.2 \text{ (s)} = 0.62 \text{ seg} \end{aligned}$$

Calculo de tiempo de desplazamiento entre la posición 3 a la posición 4 (186mm), en este desplazamiento no es posible alcanzar la velocidad máxima ya que serían necesarios dos periodos de aceleración y deceleración con una distancia de 0.131 + 0.131 m suponiendo un total de 0.262 m mayor que el desplazamiento a realizar. En este caso se calcula el tiempo transcurrido para un movimiento uniformemente acelerado durante la mitad del desplazamiento (0.186/2) a una aceleración de 3.6 m/s<sup>2</sup>.

$$t_{3 \rightarrow 4} = t_{a_{cte}}(d_t/2) + t_{-a_{cte}}(d_t/2) = 2 \times (t_{a_{cte}}(d_t/2)) = 2 \times \sqrt{\frac{2 \times (0.186(m)/2))}{3.6(m/s^2)}} = 0.44 \text{ seg}$$

El tiempo total es la suma de los tres movimientos más el de captura del vaso de palomitas:

$$t_{2 \rightarrow 3} = t_{1 \rightarrow 3} + t_{Vaso} + t_3 + t_{3 \rightarrow 4} = 0.46 + 0.2 + 0.62 + 0.44 = 1.72 \text{ segundos}$$

Por lo tanto de almacenaje deberá ser menor o igual a 1.72 segundos. Este tiempo es la suma del tiempo de cierre de la paleta, el de descenso del elevador, el de cierre de las pinzas y el de ascenso del elevador. No se suma el de apertura de las paletas que esta operación se realiza de forma paralela al ascenso del elevador.

## 4.2. Sistema de paletas de stands

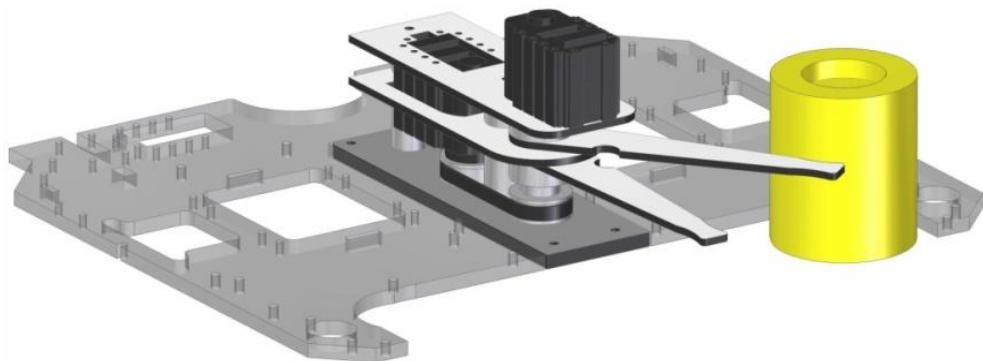


Figura 4-3 Detalle de sistema de paletas en base de robot con bloqueo de stand

Este sistema como ya ha sido comentado en varios de los capítulos precedentes dispone de dos paletas articuladas por un eje vertical y situado en el eje longitudinal de movimiento del robot. Por su disposición se trata de un sistema simétrico, pudiendo cada paleta girar 180 grados sin interferir con la otra paleta. Estas paletas en posición de captura de stands se colocan orientadas en la dirección de avance y una vez que es detectada la entrada de un stand por alguno de los sensores digitales de luz difusa E3Z-D61 de Omron colocados de forma perpendicular a la dirección de avance en cada uno de los lados de entrada (ver Figura 4-4), una de las paletas se cierra contra él inmediatamente. Este sistema permite bloquear los stands en una posición fija exacta para que el sistema de almacenaje pueda cogerlos con precisión y sin detener el movimiento del robot. Para garantizar este funcionamiento el centrado se produce exclusivamente entre tres puntos de apoyo como se puede ver en la Figura 4-4. Dos son fijos y corresponden con las varillas de guiado de los stands a lo largo de todo el recorrido de elevación y la paleta actúa como el tercer punto, habiendo sido calculado su centro de giro para que en el punto de contacto de está con el stand en la posición de bloqueo estable, haga pasar la dirección de la fuerza ejercida sobre este por el centro del stand. Se ha explicado gráficamente en la Figura 4-4.

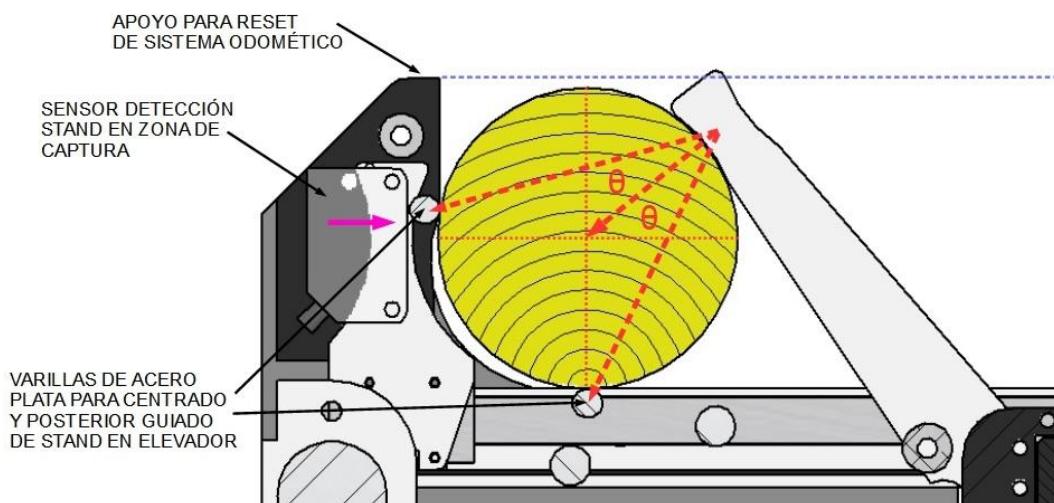


Figura 4-4 Detalle centrado stand sistema de paletas

El sistema es suficientemente rápido para evitar que los stands puedan salir despedidos en el choque contra el robot según se requería en las especificaciones. Siendo el tiempo de captura el correspondiente al giro de 38° de un servo AX12 con una velocidad de 59 rpm que resulta en 0.103 segundos. El tiempo requerido inicialmente en las especificaciones era el correspondiente a 90 grados de giro 0.3 segundos, con lo que se ha mejorado con holgura.

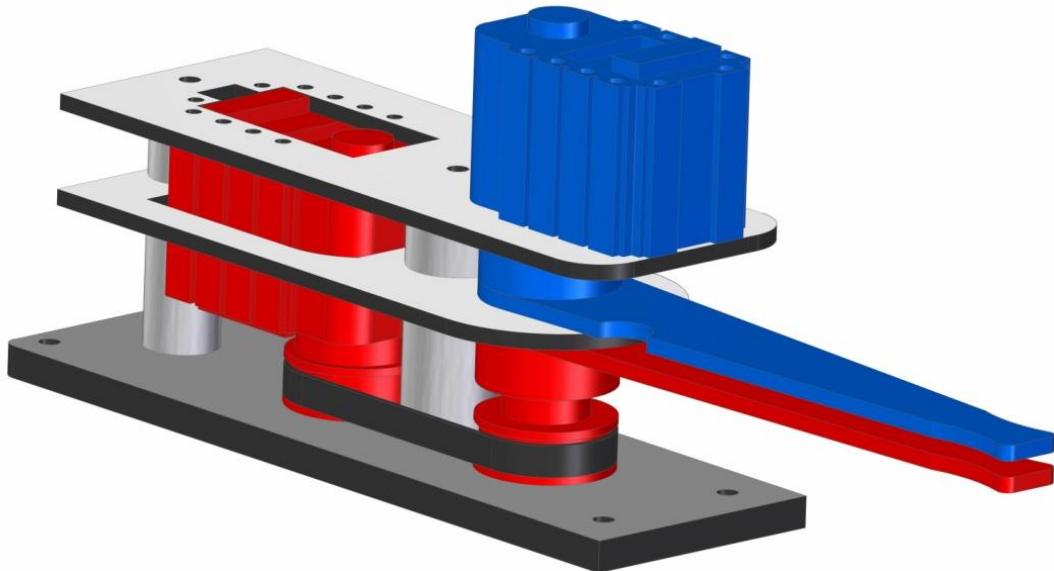


Figura 4-5 Bloque de sistema de paletas

Una de las principales características constructivas del diseño del sistema ha sido su gran compactación e independencia del resto de la estructura. La motivación para tal diseño ha sido su localización en el conjunto en una zona de muy difícil acceso, que ante una posible avería durante la competición podría causar serios problemas. El conjunto finalmente diseñado está preparado para ser retirado en bloque desde la parte inferior del robot retirando cuatro tornillos.

En la Figura 4-5 se muestra el bloque completo con los sistemas de accionamiento de ambas paletas coloreadas en diferente color para identificar claramente cada uno de sus elementos. La paleta superior de color azul esta unida mediante un separador directamente al AX12 superior, también coloreado de color azul. Sin embargo para el accionamiento de la paleta inferior de color rojo ha sido imprescindible utilizar un sistema de transmisión por correa dentada hasta su AX12 de accionamiento. Para la elección de la correa y el cálculo de la distancia entre ejes se siguió el mismo procedimiento que para las correas de transmisión del bloque de tracción del capítulo 3.3 Sistema de tracción. En este caso el módulo elegido fue de 2.5 mm al existir un requerimiento de potencia mínimo. Tampoco se ha considerado necesario estudiar el máximo de este sistema para los servos utilizados.

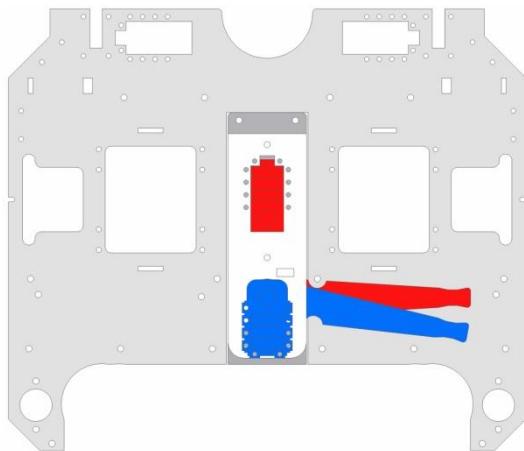


Figura 4-6 Vista en planta de sistema de paletas con posible disposición 1

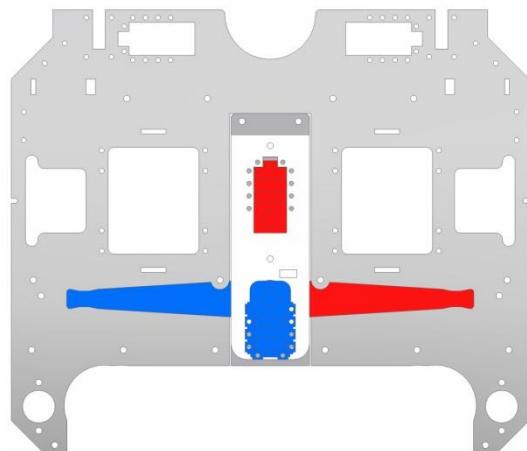


Figura 4-7 Vista en planta de sistema de paletas con posible disposición 2

En la Figura 4-6 y Figura 4-7 se han representado las posiciones máximas de desplazamiento de cada una de las paletas, estando el sistema preparado para que puedan ocultarse completamente indistintamente a uno u otro lado.

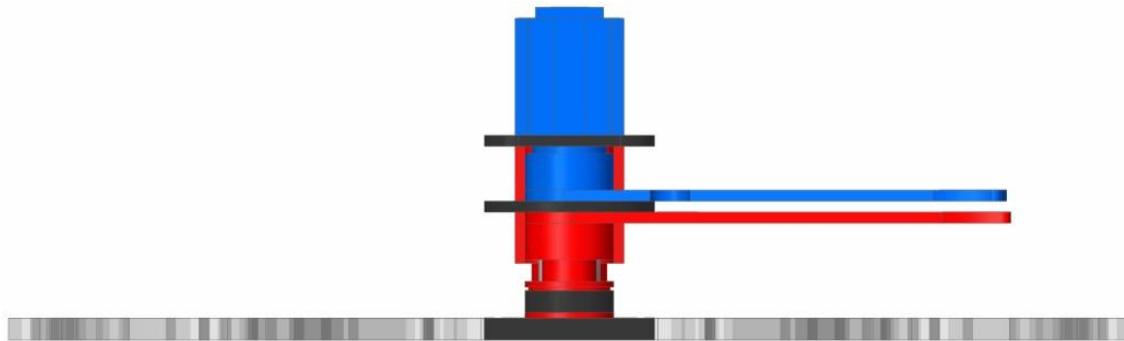


Figura 4-8 Vista frontal de sistema de paletas de stands

En la Figura 4-8 se muestra la separación existente en el plano horizontal entre las paletas que las permite girar 180 grados sin interferir con la otra.

### 4.3. Sistema de elevación de stands

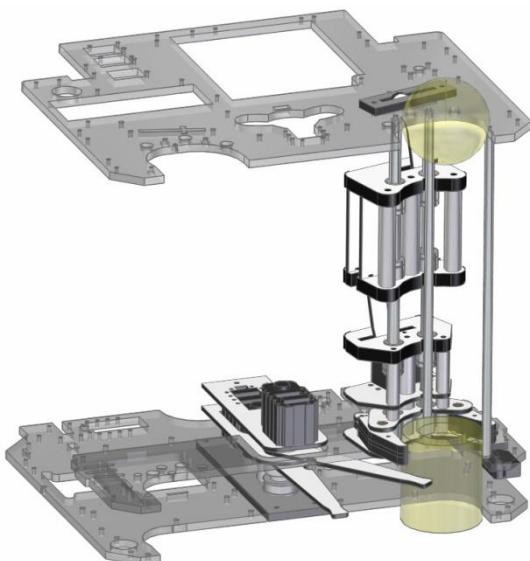


Figura 4-9 Sistema elevación derecho en posición "abajo" y sistema de paletas

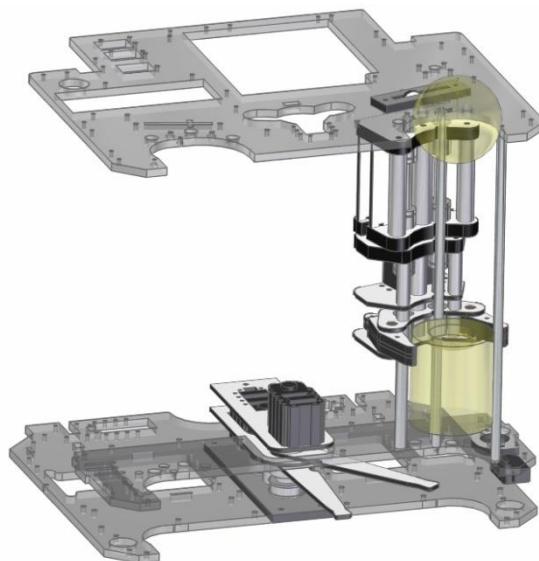


Figura 4-10 Sistema elevación derecho en posición "arriba" y sistema de paletas

El sistema de elevación está compuesto por dos conjuntos que crean un sistema que satisface todas las características que fueron concluidas en el apartado 4.1 "Prototipos y estudio previo". Se ha creado un bloque denominado "carro pinza" que se desplaza por medio de cuatro rodamientos lineales de bolas recirculantes sobre dos guías circulares de acero plata de 8 milímetros de diámetro. Este carro como su nombre indica dispone de una pinza que es capaz de agarrar los stands por su diámetro con suficiente fuerza como para evitar el deslizamiento incluso durante la construcción del foco de ocho alturas. Sobre este carro y guiado sobre las mismas guías también a través de cuatro rodamientos lineales, se ha diseñado un segundo bloque denominado "carro motor". Este carro dispone de un servo AX12 que hace girar un plato que opera a modo de cigüeñal de dos manivelas contrapuestas. Estas manivelas están unidas con dos bielas articuladas mediante ejes a la tapa superior de la estructura y al carro pinza. El diseño de las bielas como se verá más adelante permite un giro de 180 grados que producirá según él diseño una elevación exacta de 80 mm (Los stands tienen una altura de 70mm). Para entender esta explicación se puede hacer una analogía con un sistema formado por un cigüeñal con dos bielas a 180 grados y dos pistones. El conjunto se dispondría verticalmente con el eje del cigüeñal y el de los dos cilindros en el mismo eje vertical. Y tendría la particularidad de que el pistón superior tendría una posición inmóvil y serían los ejes del cigüeñal y el pistón inferior los que se desplazarían en el eje vertical.

La implementación realizada que presenta una simetría por el plano horizontal que atraviesa el eje del cigüeñal, con las manivelas y bielas de la misma longitud y en el mismo eje vertical produce un desplazamiento vertical en el carro motor exactamente de la mitad que en el carro pinza. El carro motor se desplaza 40 mm.

En la Figura 4-11 se ha esquematizado el diseño definitivo manteniendo las proporciones reales. Se han representado en color rojo las manivelas o plato ya que se trata de la misma pieza y en color verde las bielas. Los ejes de giro se han representado como puntos negros.

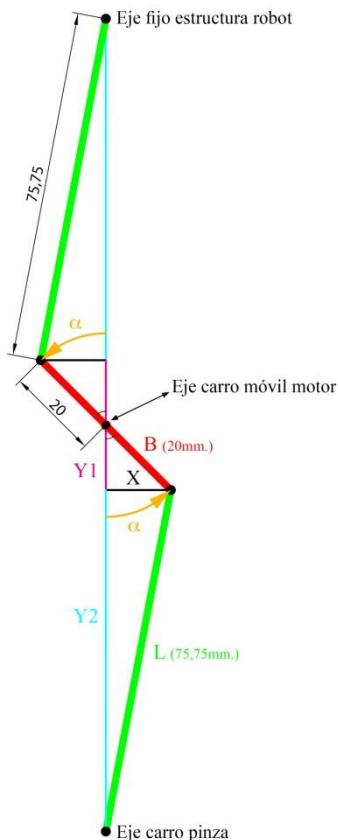


Figura 4-11 Esquematización de sistema de doble manivela-bielas con medidas de diseño definitivo

Aunque es posible mediante un simple análisis deductivo analizando las características del movimiento producido por este sistema, comprender la evolución de la velocidad y la aceleración del eje de la pinza en el eje Y, voy a plantear una resolución trigonométrica para facilitar más aún la comprensión.

En el esquema de la Figura 4-11 cada uno de los conjuntos manivela-bielas puede descomponerse en dos triángulos rectángulos. Tomando como variable de entrada del sistema el ángulo de giro  $\alpha$ , formado entre B e Y1, se puede calcular Y1, y en función de este mismo ángulo se puede deducir el cateto opuesto X. Conocido este por el teorema de Pitágoras se calcula Y2. Y por lo tanto se resuelve que la distancia entre el eje fijo de la estructura del robot y el eje del carro pinza es:

$$d_{Eje\ estructura \leftrightarrow Eje\ pinza} = 2 \times ((B \times \cos \alpha) + \sqrt{L^2 - (B^2 \times \sin \alpha)})$$

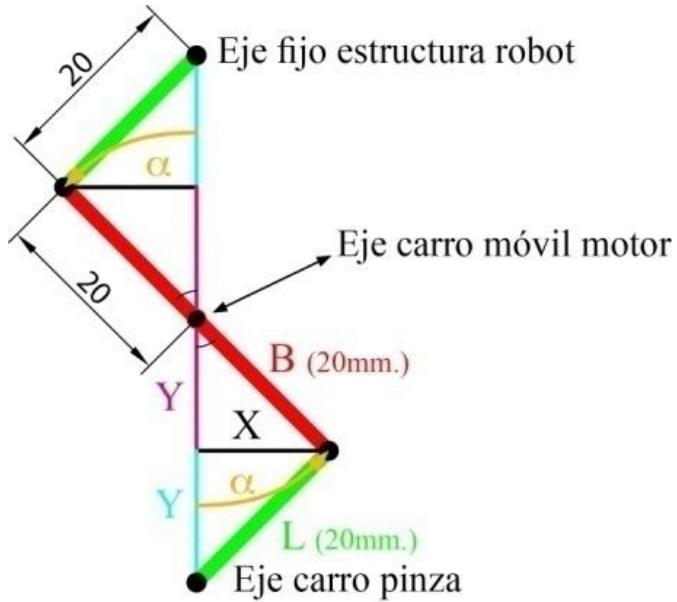


Figura 4-12 Esquematización de sistema de doble manivela-bielas con medidas para simplificación de ecuación de movimiento en el eje vertical

Como no resulta evidente la interpretación de esta ecuación se ha planteado un sistema con las bielas con la misma longitud que las manivelas para que resulte una ecuación fácilmente interpretable. La representación de este sistema se encuentra en la Figura 4-12. La ecuación resultante es la siguiente, aunque presenta el inconveniente de que solo tiene sentido mecánico hasta el giro de 90°:

$$d_{Eje\ estrucutur\ a \leftrightarrow Eje\ pinza} = 2 \times (2 \times B \times \cos \alpha)$$

En este caso el movimiento tendría una forma sinusoidal con una amplitud máxima (distancia entre ejes) de 4 veces la longitud de las manivelas y bielas. La distancia comenzaría con la amplitud máxima y a los 90° será cero.

La velocidad evolucionaría de manera inversa desde cero aumentaría hasta la amplitud máxima en los 90°.

$$v_Y = \frac{d(4B \cos(\omega t))}{d_t} = -4B\omega \sin(\omega t)$$

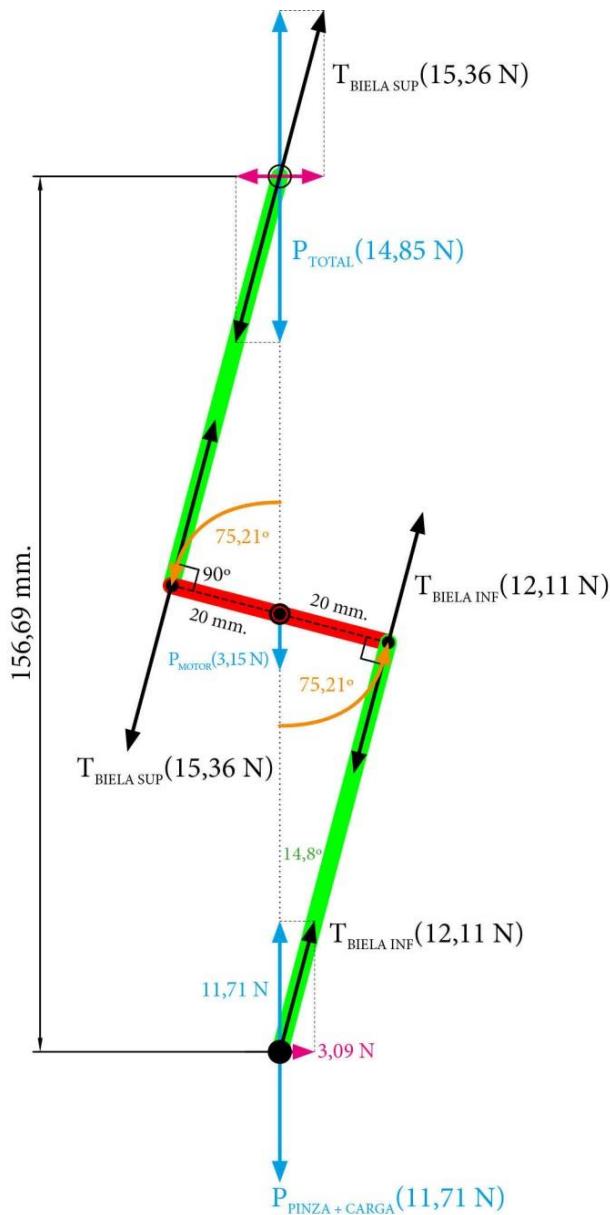


Figura 4-13 Resolución gráfica del momento resistente máximo con radio de momento en valor máximo

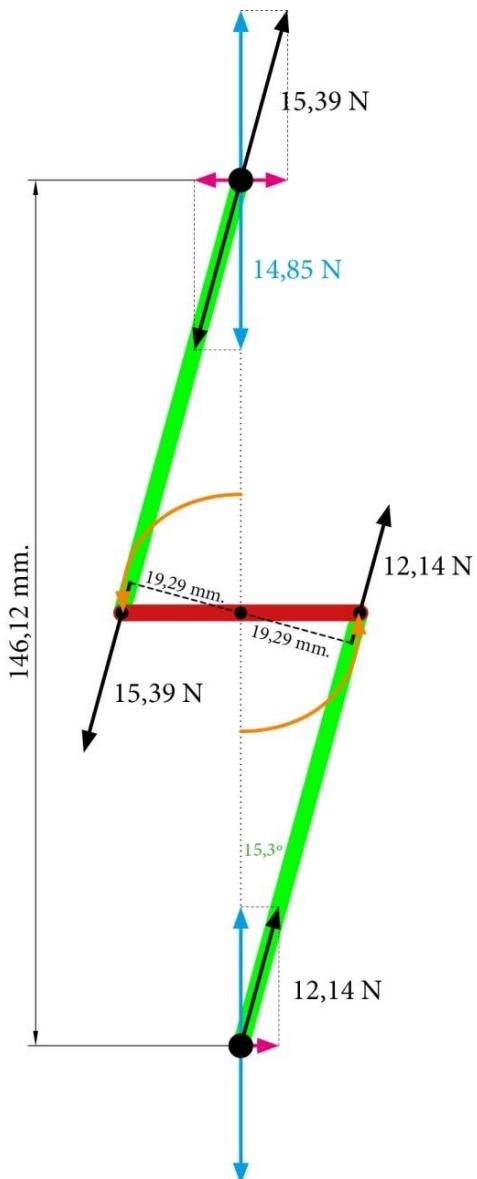


Figura 4-14 Resolución gráfica del momento resistente máximo con fuerza sobre manivela en valor máximo

En la Figura 4-13 y Figura 4-14 se ha aproximado gráficamente el momento resistente máximo durante el movimiento de elevación. Conociendo que el momento de una fuerza en cada una de las manivelas es el producto de la fuerza por la distancia más corta (la perpendicular) desde su dirección al eje de giro, el momento máximo se producirá entre los ángulos en los que se alcancen los valores máximos de variación de ambas variables y que no se dan en la misma posición. En la Figura 4-13 se representa el estado del sistema con la distancia más corta desde la dirección de la fuerza al eje de giro en su valor máximo y que lógicamente corresponde con la longitud de la manivela cuando esta se encuentra perpendicular a la biela. Al tratarse de un sistema de construcción simétrica el sistema palanca-bielas superior se encuentra en la misma situación, con la única diferencia de que el módulo de la fuerza  $T_{BIELA\ SUP}$  resultante aplicada sobre la manivela tiene un valor superior

obtenido gráficamente como puede verse en la Figura 4-13 a partir de  $P_{TOTAL}$  en el que ha sido incluido el peso de todos los elementos del sistema  $P_{PINZA+CARGA} + P_{MOTOR}$ . El momento resultante se obtiene con la suma de los dos momentos de ambas manivelas.

$$\begin{aligned} M_{Resistente\_MAX} &= M_{Manivela\ INF} + M_{Manivela\ SUP} = T_{Biel\ INF} \times L_{Manivela\ INF} + T_{Biel\ SUP} \times L_{Manivela\ SUP} \\ &= 12.11N \times 0.02m + 15.36N \times 0.02m = 0.242N + 0.307 = 0.549N \text{ a } 75.21^\circ \end{aligned}$$

En la Figura 4-14 se representa el estado del sistema con el valor de las fuerzas que actúan sobre las manivelas en su valor máximo. Como se puede ver gráficamente con  $P_{PINZA+CARGA}$  constante, la fuerza  $T_{BIELA\ INF}$  aumenta progresivamente con el aumento del ángulo de apertura entre esta última fuerza y el eje vertical, alcanzándose el máximo aumento cuando la manivela se encuentra en posición horizontal. Siendo el momento resultante obtenido.

$$M_{Resistente\_MAX} = (12.14N + 15.39N) \times 0.01929m = 0.531Nm \text{ a } 90^\circ$$

De estos dos cálculos se deduce que el momento máximo será superior a  $0.549Nm$  en un ángulo de giro entre  $75.21^\circ$  y  $90^\circ$ . En cualquier caso, y siempre teniendo en cuenta que no se han tenido en cuenta ningún tipo de pérdidas por rozamiento, este valor nos da una aproximación suficiente para valorar que un servo AX12 con un par máximo de  $1.52Nm$  como poco será capaz de elevar la carga.

La hoja de características de este servo carece de los datos necesarios para poder calcular la velocidad del movimiento, además una de las características de este servo es el ajuste de la velocidad de giro, pero siempre suponiendo que nos encontraremos por debajo de su potencia mecánica máxima que no es conocida.

Para desplazar el sistema diseñado a la velocidad de giro máxima del servo aproximadamente  $60 \text{ rpm}$  ( $2\pi \text{ rad/s}$ ) la potencia mecánica necesaria sería de.

$$P(w) = M(Nm) \times \omega(rad/s) = 0.531Nm \times 2\pi(rad/s) = 3.33w$$

El único dato de la hoja de características que permite hacer una aproximación a la potencia mecánica es la limitación de corriente máxima que asciende a  $0.9A$ , que con la tensión a la que van a ser alimentados de  $12V$ , resultaría en una potencia eléctrica máxima de  $10.8w$ . Aunque podría parecer suficiente para permitir al servo moverse a una velocidad próxima a su máxima al desconocer los valores de rendimiento no es posible llegar a una conclusión determinante.

Para completar el estudio del sistema se ha hecho uso de un software específico de simulación de mecanismos que va a permitir con gran facilidad graficar todos los resultados de las diferentes ecuaciones de movimiento planteadas.

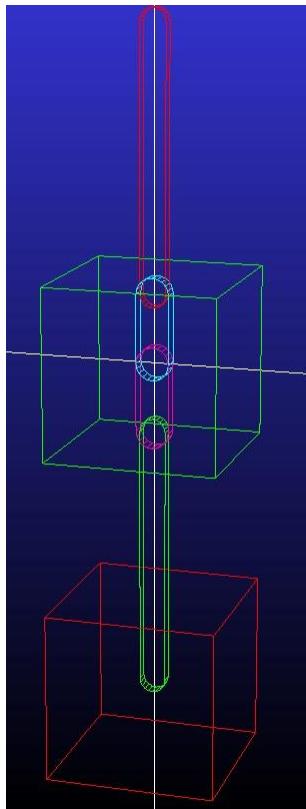


Figura 4-15 Sistema implementado para estudio de sistema de elevación en MS Adams  
Vista 3D

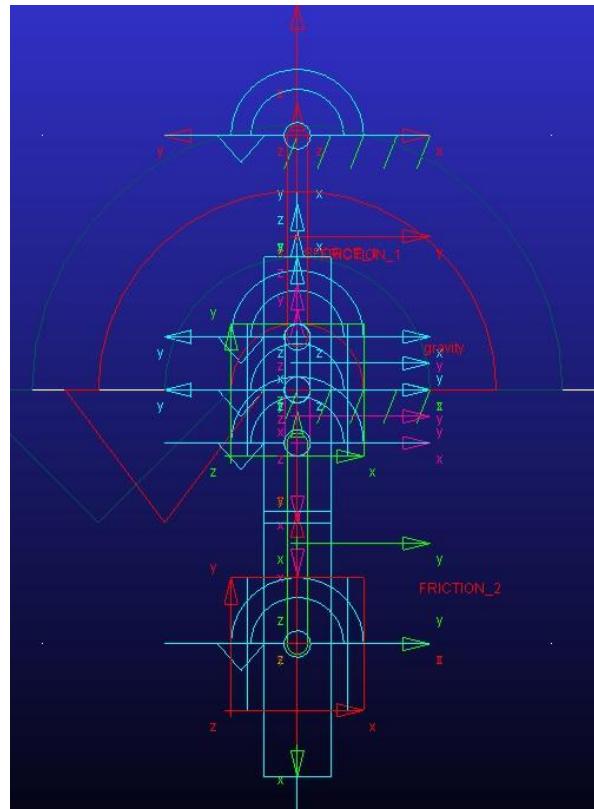


Figura 4-16 Sistema implementado para estudio de sistema de elevación en MS Adams con representación de todos los elementos incluidos (articulaciones, guías, etc.)

Se ha utilizado el software MS ADAMS en su versión de estudiante, creando un sistema con el mecanismo simplificado. Este programa permite por un lado hacer simulaciones más orientadas al estudio cinemático mediante la aplicación de movimientos en sus diferentes elementos, o mediante la aplicación de fuerzas o momentos de fuerzas sobre ejes para realizar estudios con un enfoque de la dinámica del movimiento. En la Figura 4-15 y Figura 4-16 se muestra una captura de pantalla del sistema implementado para su estudio de forma simplificada con vista 3D y con todos los elementos incluidos respectivamente. Los principales elementos han sido la articulación superior fija, los bloques de los dos carros con sus respectivas guías verticales y el resto de articulaciones y bielas. Aunque es posible una vez creada la geometría en 3D que puede ser exportada desde otras aplicaciones de CAD, aplicar masa a cada uno de los elementos, como en este caso pueden ser las bielas, se ha preferido inicialmente plantear un sistema tan simplificado como el utilizado durante los cálculos previos para que sirva de comprobación de estos últimos.

La primera simulación realizada ha sido puramente cinemática utilizando las siguientes variables:

- Peso de carro pinza 1.1948Kg y carro motor 0.321Kg.
- Función de movimiento constante sobre el eje motor de 59 rpm "354d × time".
- Tiempo de simulación 0.5084s para completar un movimiento completo de 180°.
- Inicio de movimiento en posición inferior y final en posición superior.

En la gráfica de la Figura 4-17 se muestra el resultado obtenido para la evolución de la posición, la velocidad y la aceleración en el eje vertical del punto de articulación del carro pinza.

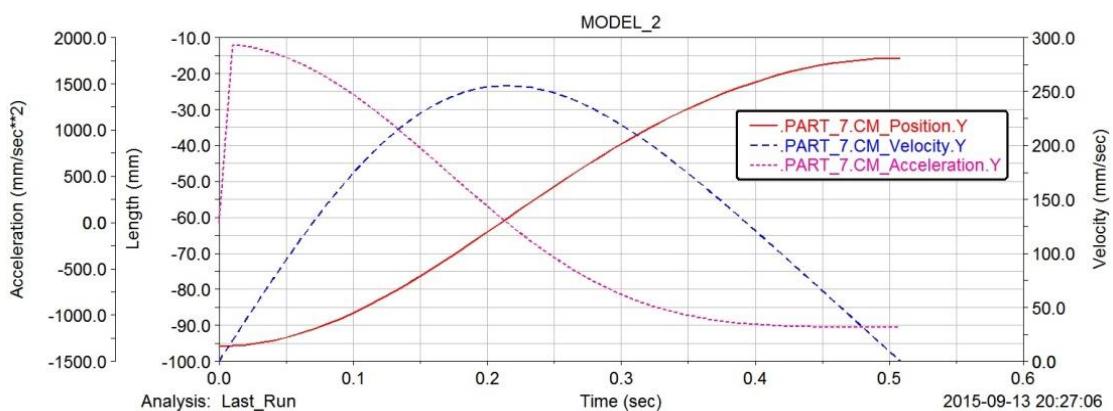


Figura 4-17 Grafica MS Adams cinemática carro pinza en eje Y

Lo más destacable de esta grafica además de los valores máximos alcanzados, es la variación de la velocidad con la forma aproximada a una onda sinusoidal que se adelantaba en los estudios previos y que permite de una forma sencilla conseguir las rampas de aceleración y deceleración necesarias para evitar la caída de la torre durante su construcción.

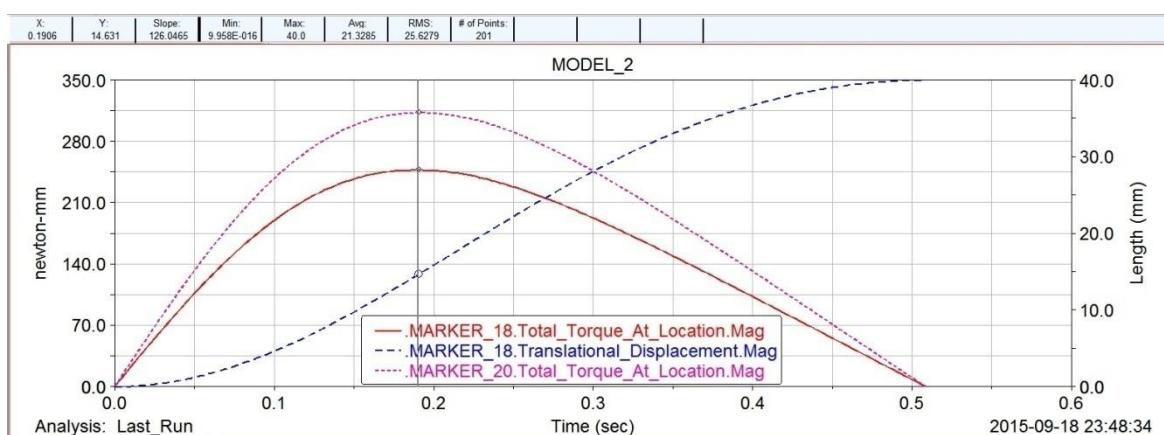


Figura 4-18 Momento máximo sobre manivelas y coordenada Y de eje manivela-bielas inferiores.

En la gráfica de la Figura 4-18 se representa la evolución del momento de la fuerza sobre la manivela inferior en línea roja, y sobre la manivela superior en línea punteada rosa y el desplazamiento vertical del eje de unión de la manivela inferior con su biela respecto al eje de giro del servo. La línea vertical marca el punto en el que se alcanzan los máximos momentos sobre ambas manivelas y a la altura de referencia a la que se producen ambos al mismo tiempo según se ha comprobado. Estos son respectivamente de 247.47 Nmm en la manivela inferior y de 312.99 Nmm para la superior a la altura de referencia tomada de 14.63 mm. Se ha calculado y esta cota de elevación corresponde con  $74.42^\circ$ . Por lo tanto el momento máximo que corresponderá a la suma de ambos es de 560.46 Nmm y se produce un grado antes de alcanzar la perpendicularidad entre las manivelas y las bielas en el sentido de movimiento ascendente. El valor que se approximó gráficamente fue de 0.549 Nm con lo que se puede considerar correcto el sistema utilizado.

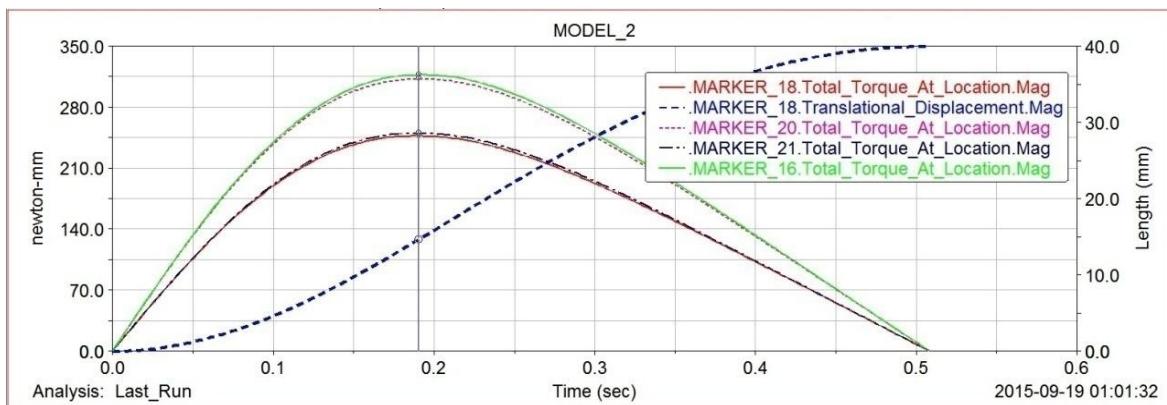


Figura 4-19 Momento máximo sobre manivelas con rozamiento dinámico en guías de 0.05.

En la gráfica de la Figura 4-19 aparecen sobre las curvas de la representación anterior las correspondientes a una nueva simulación incluyendo un coeficiente de rozamiento estático y dinámico sobre las dos guías de los carros con un valor 10 veces superior a un valor típico de este tipo de guías que no suele superar 0.005 (el valor utilizado ha sido de 0.05). Los valores de los momentos han sido de 250.62Nmm y 317.6Nmm dando un momento total de 568.22Nmm, que parece un valor despreciable, para lo que hay que tener en cuenta que el sistema simulado tiene un diseño de disposición de centro de masas completamente idealizado, apareciendo únicamente fuerzas radiales sobre la guía del carro motor como consecuencia del momento contrario que se opone al generado por el servo para la elevación del sistema. En la práctica este valor se ha comprobado que ha sido mucho mayor no solo por las fuerzas radiales despreciadas por las disposiciones de los centros de masas, sino principalmente por la complejidad de la construcción de un sistema de dos guías que deben disponerse completamente paralelas para evitar fuerzas radiales en los ejes que pueden ser muy importantes en comparación a las anteriormente comentadas.

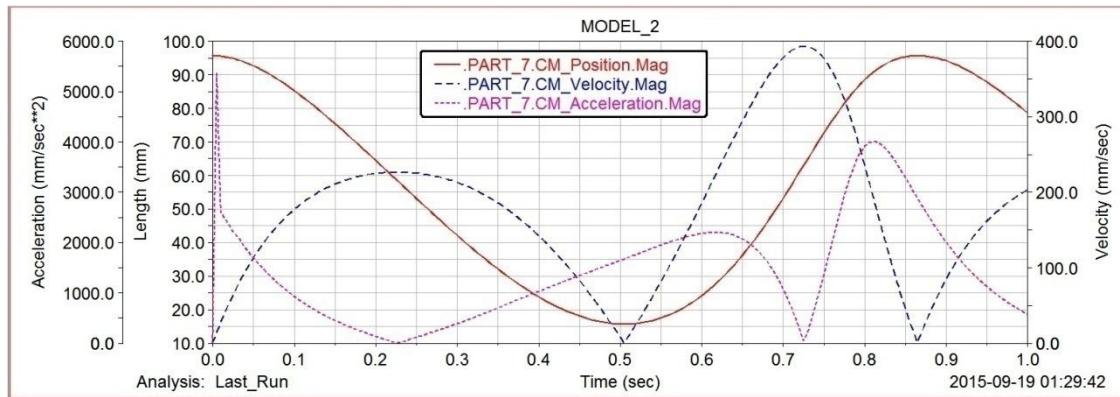


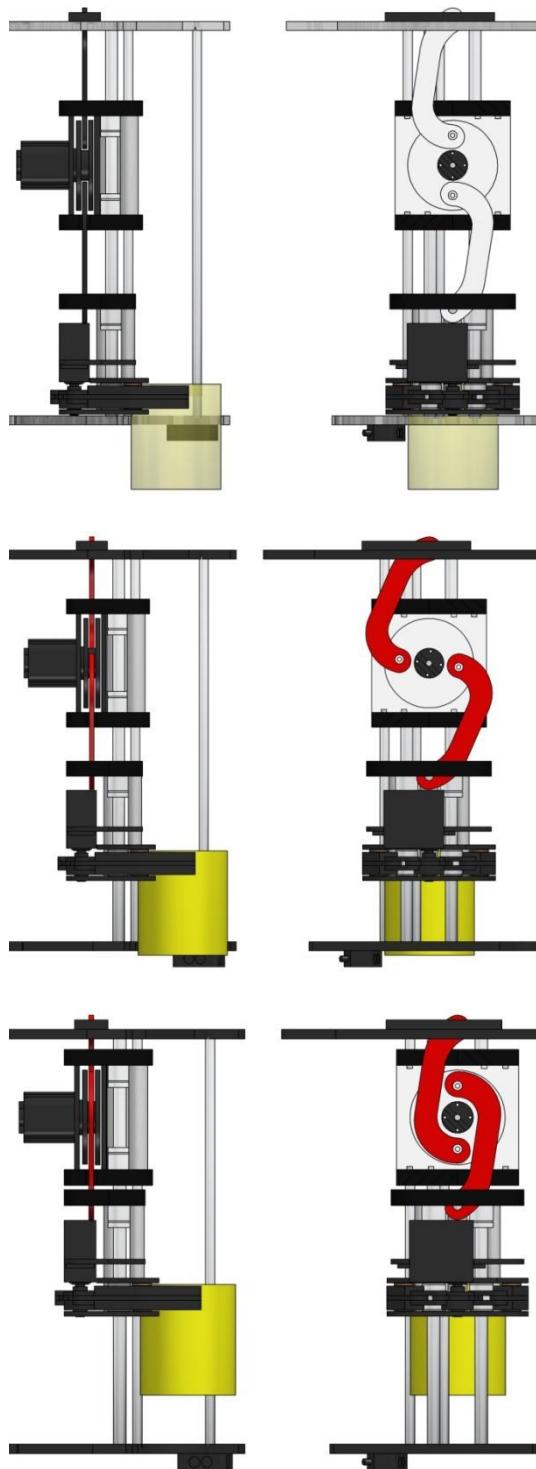
Figura 4-20 Grafica cinemática carro pinza en eje Y par motor en función de rad/s

Por último aunque no se conoce la curva del par motor se ha hecho una simulación dinámica del sistema con un torque en función de la velocidad angular de giro del servo. La función introducida ha sido la siguiente.

$$T_{AX12} = 2059.77 \text{ (Nm)} - (373.058 \left( \frac{\text{Nmm}}{\text{rad/s}} \right)) \times \omega_{AX12}(\text{rad/s})$$

Se ha hecho una simulación de un segundo de movimiento para ver la evolución de las gráficas, una vez comenzado el movimiento descendente. En esta grafica se comprueba que si el motor presentara el comportamiento de esta recta torque velocidad angular el motor alcanzaría la cota superior en el medio segundo esperado.

En esta secuencia de imágenes se muestra gráficamente el funcionamiento del sistema de bielas, y el diseño de estas que permite un movimiento de  $180^\circ$  encontrándose ambas en el mismo plano.



Carro pinza en cota vertical mínima (0 mm) con el sistema en equilibrio, presentando la carga un momento de fuerza nulo sobre el eje del motor.

El carro ha de bajar a esta posición para recoger los stands bloqueados por el sistema de paletas.

Carro pinza a mitad del recorrido de elevación (40mm). Este no se corresponde con la mitad del recorrido de giro.

Posición próxima a la de mayor momento resistente al de elevación.

Carro pinza en cota vertical de máxima elevación (80 mm) con el sistema en equilibrio, presentando la carga un momento de fuerza nulo sobre el eje del motor.

Figura 4-21 Secuencia elevación

#### 4.3.1. Carro pinza



Figura 4-22 *Vista delantera de carro pinza izquierdo*

Figura 4-23 *Vista trasera de carro pinza izquierdo*

En el diseño de este bloque se pueden distinguir dos partes claramente diferenciadas. Por un lado el diseño de la estructura del bloque con la disposición de los diferentes elementos y sus centros de masas respecto a las dos varillas de guiado, que como se va a explicar a seguido unos criterios para la reducción de la magnitud de las fuerzas de rozamiento y por otro el propio sistema utilizado para accionar las pinzas con la suficiente fuerza como para evitar el deslizamiento de los stands durante la elevación. En la Figura 4-22 se muestra una vista delantera del carro pinza donde se puede apreciar la disposición de los cuatro rodamientos lineales y el sistema multicapa tipo "sandwich" utilizado para crear las diferentes estructuras de soporte para los ejes de giro, los propios rodamientos y el servo utilizado para generación del movimiento. Estas capas estás separadas por un sistema de casquillos de aluminio autocentrables en disposición de triángulo y unidos mediante unos tornillos que los atraviesan. Este sistema como seguramente ya se ha observado hay sido utilizado frecuentemente en gran parte de las estructuras creadas, por la simplicidad de su diseño y sobre todo dado el sistema de fabricación utilizado la mayor precisión que aporta en el resultado final frente a otros sistemas con uniones en escuadra, uniones pegadas, soldadas o incluso con piezas de chapa plegadas. Y en este sistema hay que recordar que era de crucial importancia la disposición de los 4 rodamientos perfectamente alineados y paralelos. En la Figura 4-23 se muestra una vista trasera del carro pinza pudiéndose observar la disposición del servo y el sistema de manivelas y bielas utilizado para articular el movimiento de las pinzas.

Para el diseño de la estructura y la disposición de los diferentes elementos dentro de esta, se ha primado por un lado lograr la máxima compresión posible del sistema y por otro minimizar las perdidas por rozamiento por fuerzas radiales en los rodamientos.

En un sistema ideal que no produciría ningún tipo de fuerza radial sobre los rodamientos lineales, todos los centros de masas o él resultante de todos ellos, se encontraría en el centro geométrico exacto entre las dos guías de desplazamiento, en donde también se encontraría en punto de tracción de la biela. Siendo este el objetivo final durante el diseño, el espacio limitado para este y otras consideraciones de diseño han resultado en la disposición de la vista inferior en planta de la Figura 4-24. Aunque se ha conseguido que los centros de masas se encuentren alineados en el eje vertical a la misma distancia de ambas guías el centro de masas de la carga se encuentra verticalmente a 43 mm por debajo de las guías y el punto de tracción de la biela a 18 sobre estas.

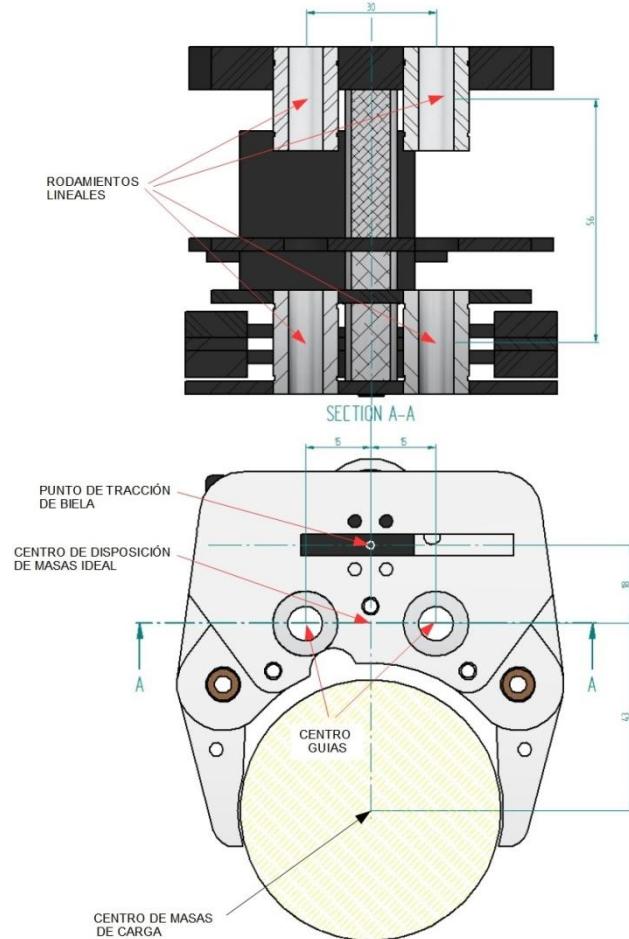


Figura 4-24 Detalle disposición de elementos en diseño de carro pinza

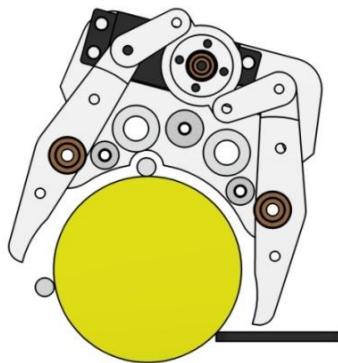
Una vez concluida la disposición en planta de los diferentes elementos y conociendo la existencia inevitable de unas fuerzas radiales debidas a esta disposición sobre los rodamientos, que aumentaran las fuerzas de rozamiento y reducirán el rendimiento, se ha concluido el diseño intentando reducir al máximo las componentes de estas fuerzas radiales. Esto se consigue aumentando el distanciamiento vertical al máximo posible entre los rodamientos lineales. La altura final del bloque y el distanciamiento de estos finalmente ha sido limitada por el espacio máximo disponible con el sistema de elevación completo en la posición de elevación máxima. En este punto del movimiento el espacio es el mínimo disponible y se ha dejado una holgura entre los diferentes carros tan solo de 3 mm.

El sistema de accionamiento de la pinza utilizado se basa en el mismo principio que el diseñado para el sistema de elevación, pero con la diferencia de que en este caso el eje motor es fijo y el desplazamiento lineal generado con el conjunto manivela-bielas a ambos lados de

este es completamente simétrico. En este caso la principal ventaja de este sistema no radica en la reducción de la velocidad de avance lineal en las posiciones de equilibrio (cuando los tres ejes se encuentran alineados), sino en la reducción del momento resistente sobre el eje motor al alcanzar esta posición, ya que la mínima distancia perpendicular desde la dirección de la fuerza al eje de giro se reduce a medida que se alinean ambos ejes y por lo tanto en la misma medida el momento de la fuerza sobre dicho eje. Una vez alineados los ejes el momento sobre el eje es nulo por lo que con el diseño utilizado una vez agarrado el stand el servo no se encuentra bajo ninguna carga siendo su consumo mínimo. Estas características han permitido utilizar para el accionamiento un servo estándar Futaba 3003S. La hoja de características de este servo se encuentra en la ruta del soporte digital *DOCUMENTOS TÉCNICOS\ELECTRONICA\SERVOS\Futaba 3003\Futaba-s3003.pdf*.

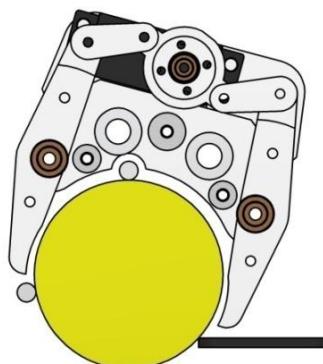
La longitud de las manivelas, bielas y pinzas ha sido calculada para que entre la posición de apertura máxima (necesaria para liberar el foco con un movimiento de retroceso del robot) y la de bloqueo se realice un giro de 45°, que con el tipo de servo utilizado con una velocidad de 0.00383 ( $s/\text{deg}$ ) establecería el tiempo de operación en 0.17 seg.

A continuación se muestra la secuencia de la articulación del movimiento de cierre del sistema de la pinza.



En la Figura 4-25 la pinza se encuentra en la posición de apertura máxima, no existiendo ninguna fuerza aplicada sobre los extremos de las pinza. Este punto coincide con el de máxima velocidad tangencial en el extremo de la pinza y mayor momento resistente transmitido al eje del servo desde esta.

Figura 4-25 Sistema pinza en posición de apertura máxima (giro 45°)



En la Figura 4-26 con un ángulo de giro recorrido de 22.5° se observa como los ejes de las bielas se van alineando progresivamente reduciendo del momento de la fuerza resistente transmitida al eje del motor, según se ha explicado en la página anterior.

Figura 4-26 Sistema pinza en posición de apertura media (giro 22.5°)

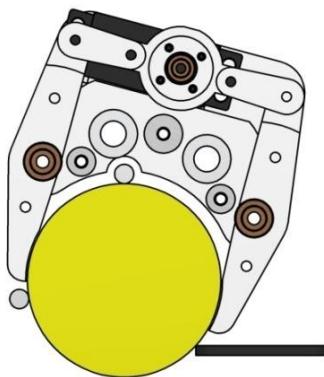


Figura 4-27 Sistema pinza en posición cerrada  
(giro 0°)

En la Figura 4-27 el sistema ha alcanzado su posición de equilibrio con un momento nulo sobre el eje del servo. Para asegurar que el diámetro variable de los stands según la tolerancia indicada en la normativa de Eurobot (Eurobot, 2014), pueda evitar alcanzar esta posición se ha dado una holgura entre en diámetro del apoyo de la pinza y el stand de 0.5 mm que es suplementado con una banda de goma siliconada de 1 mm de espesor.

En la Figura 4-28 se muestra una explosión del conjunto del carro pinza izquierdo donde se puede observar el sistema tipo "sandwich" que se comentó en la introducción. También aparecen visibles los casquillos de bronce sinterizado (en color dorado) utilizados para minimizar las perdidas por rozamiento en los ejes de las pinzas y del punto de apoyo opuesto del eje del servo de accionamiento.



Figura 4-28 Detalle constructivo  
carro pinza izquierdo Explosión

#### 4.3.2. Carro motor



Figura 4-29 Vista delantera de carro motor izquierdo



Figura 4-30 Vista lateral de carro motor izquierdo

La sistemática utilizada para el diseño de este bloque ha seguido los mismos principios que los desarrollados para el carro pinza. En la parte estructural ha primado la reducción de las fuerzas radiales sobre los rodamientos lineales pero en este caso con la particularidad de que el principal causante de estas no es la distribución de masas del conjunto respecto al eje que une ambas guías, si no las fuerzas torsionales opuestas a las creadas por los momentos de fuerzas sobre el eje del servo. Lógicamente si la disposición de los rodamientos no bloqueara el giro del carro motor sobre el eje del servo, este giraría en sentido opuesto al que produce el movimiento de elevación. La solución al igual que en el carro pinza ha pasado por aumentar la separación vertical de los rodamientos lo máximo que ha permitido el diseño del resto del sistema de elevación.

Este bloque tenía la particularidad de la presencia de las paredes de sujeción del sistema del plato del servo colocadas verticalmente y que ha sido resuelto sin ningún tipo de fijación a través del sistema de pestañas ya utilizado en el ensamblado del resto de la estructura. En la Figura 4-31 se pueden observar dichas pestañas y el sistema empleado. Este conjunto se completa con el subconjunto del plato servo que está fijado a la cabeza de este a través de cuatro tornillos y conforma el sistema de manivelas contrapuestas 180°.



Figura 4-31 *Detalle ensamblado de carro motor por pestañas*



Figura 4-32 *Conjunto de sistema de manivelas por plato de servo*

Por último es importante indicar que todos los ejes articulados del conjunto del sistema de elevación tienen una de las articulaciones encasquilladas con bronce sinterizado. Siendo en el caso de las uniones "plato-manivelas" a bielas, las bielas las que tienen estos casquillos.

#### **4.4. Sistema carro horizontal de stands**

---

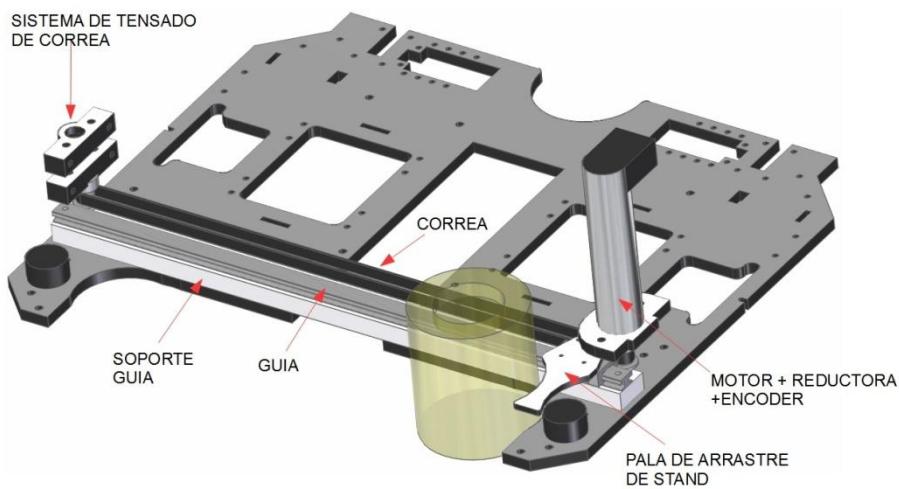


Figura 4-33 *Conjunto elementos carro horizontal sobre base robot*

Este sistema permite desplazar el stand desde una posición de bloqueo de paleta a la contraria o viceversa. Permite intercambiar los stands entre los almacenes verticales para la construcción del foco de 8 niveles de apilamiento. En la Figura 4-33 se muestra todos los

elementos de este sistema en su ubicación sobre la base del robot, también se ha incluido un stand semi-trasparente en la posición de bloqueo derecha.

Se ha diseñado para que pueda tener un comportamiento completamente simétrico pudiéndose referenciar a ambos lados y mover stands desde cualquiera de las posiciones de bloqueo a la contraria, en la Figura 4-33 se puede ver como la paleta utilizada para el desplazamiento tiene una cuna con forma circular simétrica. La longitud del desplazamiento total ha obligado a accionarlo a través del arrastre de una correa dispuesta entre dos poleas, paralela a la guía utilizada. Se ha utilizado una guía comercial de aluminio tratado con un patín de plástico autolubricado de la marca Igus.

La elección de las poleas utilizadas debido a mínimo espacio disponible para la implementación del sistema se ha convertido en el punto de partida del diseño del sistema de accionamiento. Para que el carro que soporta la pala que desplaza los stands pueda desplazarse los 278 mm de carrera entre topes mecánicos y pueda quedar oculta en los laterales del robot ha sido necesario recurrir a las poleas dentadas de menor diámetro que son distribuidas. En concreto se han elegido dos poleas dentadas de 12 dientes con paso métrico de 2.5mm y 9.6 mm de diámetro primitivo. El módulo elegido no va a comprometer la aplicación ya que la potencia mecánica para desplazar el stand horizontalmente es despreciable. Por otro lado aunque se han utilizado los mismos cálculos empleados en el apartado 3.3 "Sistema de tracción" para el cálculo de la distancia entre ejes, se ha optado en este caso debido a la longitud de la correa por incorporar un sistema de tensado mediante un sistema de desplazamiento y fijación del eje de la polea loca (ver Figura 4-33). Por último el motor utilizado en este caso no va a ser elegido para adaptarlo a las especificaciones de diseño sino que como otros ha venido impuesto como una especificación para reducir los costes finales del proyecto, reutilizando elementos recuperados de anteriores diseños. Por supuesto se ha comprobado que se encontraba dentro de un rango aceptable para el accionamiento de este sistema. A continuación se detallan las principales características del motor utilizado.

- Motor Maxon 2322.980-52.235-200
- 12V-6W
- Reductora 14:1
- Revoluciones nominales antes de reductora: 4300 rpm
- Revoluciones sin carga: 7010 rpm

Aproximación a la velocidad del carro horizontal máxima tomando como velocidad de giro del motor la media de su rango máximo.

$$\left( \frac{7010 - 4300}{2} \right) + 4300 = 5644 \text{ rpm}$$

$$i_{Reducción} = \frac{RPM_{MOTOR}}{RPM_{POLEA}}; RPM_{POLEA} = \frac{RPM_{MOTOR}}{i_{Reducción}} = \frac{5644 \text{ (rpm)}}{14} = 403.14 \text{ rpm}$$

$$\begin{aligned} v_{Carro} &= \omega \times r_{POLEA} = 403.14 \left( \frac{rev}{min} \right) \times 2\pi \left( \frac{rad}{rev} \right) \times 60^{-1} \left( \frac{min}{seg} \right) \times \left( \frac{0.0096}{2} \right) (m) \\ &= 0.202 m/s \end{aligned}$$

El tiempo en alcanzar la mitad del recorrido es.

$$t_{centro} = \frac{d_{Total}/2}{v} = \frac{0.278(m)/2}{0.202(m/s)} = 0.68 \text{ seg}$$

Se ha calculado el tiempo aproximado solo hasta la mitad del recorrido por que como se va a detallar más adelante para optimizar el tiempo de construcción del foco, el carro solo arrastrara el stand hasta este punto pasando a retroceder de inmediato a por otro stand, siendo las paletas las encargadas de completar el movimiento y bloquear el stand en la posición de elevación del lado opuesto.



Figura 4-34 *Carro horizontal en mitad de recorrido*

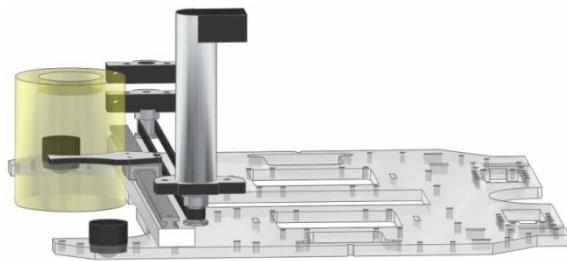


Figura 4-35 *Carro horizontal en mitad de recorrido vista lateral*

Por último se ha incluido la Figura 4-36 con la representación de la explosión de todos los elementos del sistema para facilitar su comprensión.

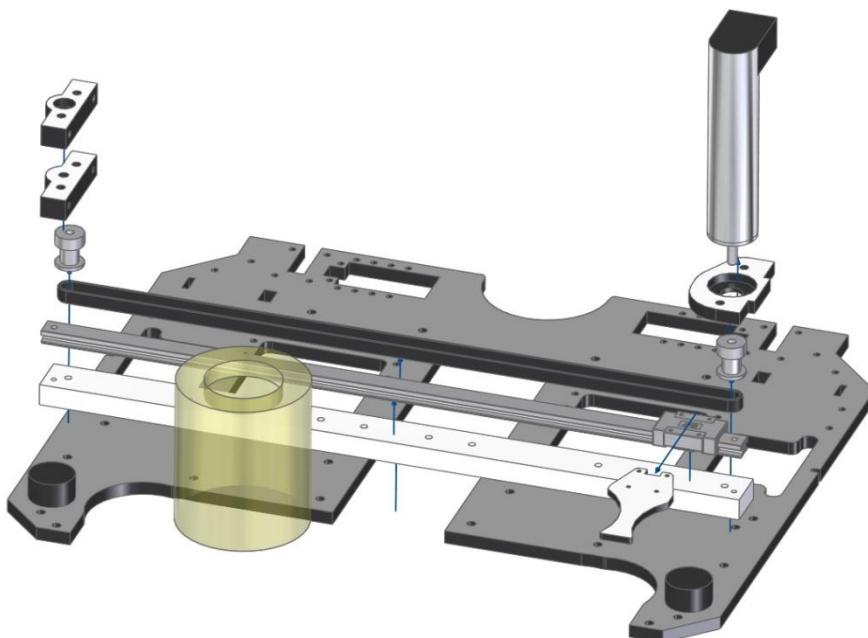


Figura 4-36 *Detalle elementos sistema carro horizontal explosionados*

Como se puede ver no se ha integrado ningún tipo de sensor de final de carrera en los extremos de los desplazamientos, ni para evitar choques ni para utilizarlos como sensores de referencia para el sistema de control de encoder incremental. Se ha optado por simplificar la

construcción y reducir el número de entradas en la tarjeta de control por realizar la operación de referenciado por control software. Mediante funciones implementadas en las librerías que van a ser utilizadas como base en la fase de programación, se detectara el bloqueo del motor al enviarlo a una consigna de posición mayor que la distancia máxima de desplazamiento, lo que producirá el choque inevitablemente con el final de carrera mecánico. Una vez detectado el bloqueo se le asignara a la variable de programa de posición el valor correspondiente, y se evitaran choques estableciendo límites software que no podrán ser sobrepasados.

#### 4.5. Sistema de puertas de stands

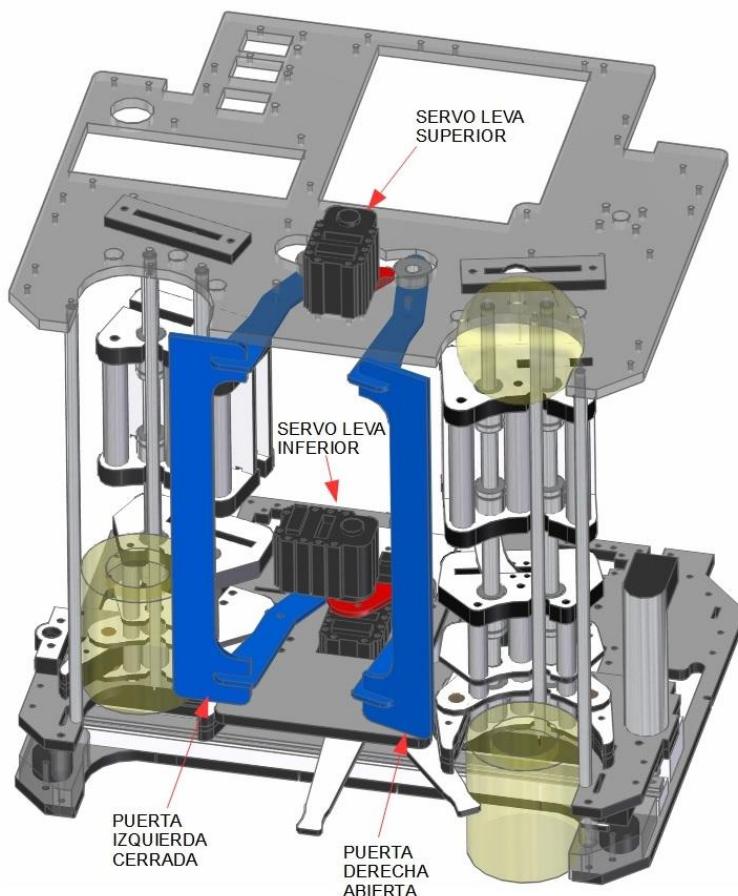


Figura 4-37 Sistema de puertas de stands integrado en sistema de manipulación de stands

Como se ha adelantado este sistema tiene una doble función. Por un lado es uno de los tres puntos de guiado de los stands en el almacén vertical, los otros dos son fijos (ver Figura 4-37) y por otro mediante su articulación permite la liberación de los stands apilados. Este sistema es doble habiéndose implementado uno para cada uno de los almacenes pero ha sido diseñado para ser accionado por el movimiento de una única leva que dependiendo del lado hacia el que gire acciona una u otra puerta, y por lo tanto esto imposibilita la apertura de ambas al mismo tiempo. Aunque el sistema es el explicado se tomó la decisión de clonarlo con un funcionamiento idéntico en los brazos de articulación inferiores y superiores de las puertas para evitar que la posible falta de rigidez en la construcción de estas pudiera comprometer el ángulo de apertura del brazo de la articulación inferior en el caso de situar

una única leva en el brazo de articulación superior. El cierre de las puertas se mantiene con dos cordones elásticos por puerta que traccionan de los extremos de los brazos de articulación superior e inferior en sentido contrario a la fuerza de apertura producida por la leva. La dirección de tracción de estos cordones se ha representado en la Figura 4-38. También se pueden ver los rodamientos sobre los que actúan las levadas para minimizar las perdidas por rozamiento.

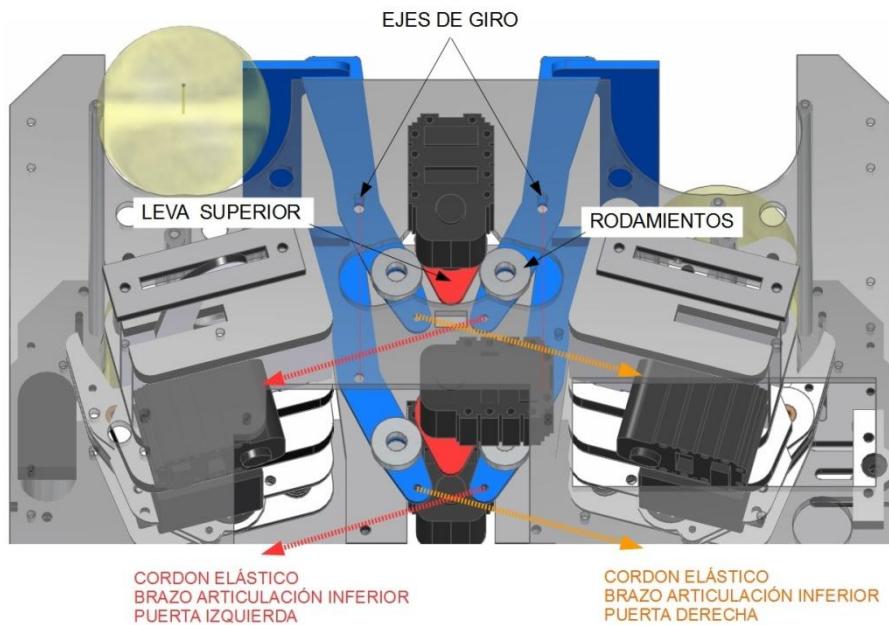


Figura 4-38 Sistema puertas de stands en posición cerradas con detalle de cordones elásticos de cierre.

Se ha descartado realizar ningún tipo de cálculo de dimensionamiento de los servos ya que para simplificar el control han sido utilizados AX12, estando estos ampliamente sobredimensionados para este accionamiento.

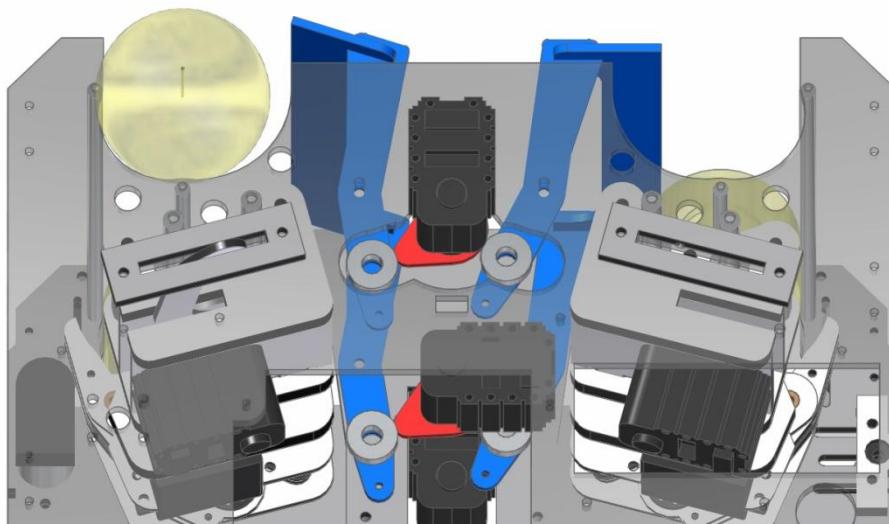


Figura 4-39 Sistema puertas de stands con puerta derecha abierta.

## 4.6. Secuencia de construcción de foco

En este apartado se va a describir la secuencia completa de movimientos de la operación de construcción del foco de ocho alturas. Por su puesto ha de construirse en el almacén donde este precargada previamente la pelota de tenis.

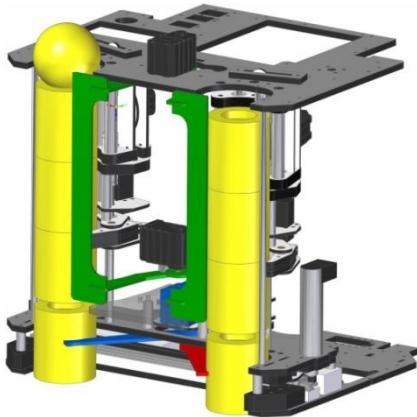


Figura 4-40 Secuencia foco 1



Figura 4-41 Secuencia foco 2

1. La secuencia comienza en el estado de la última recolección. Los dos stands inferiores se encuentran bloqueados por su respectiva paleta y las pinzas permanecen arriba sujetando los últimos stands elevados.
2. Mientras que las paletas continúan sujetando los stands inferiores las pinzas descienden y agarran dichos stands.

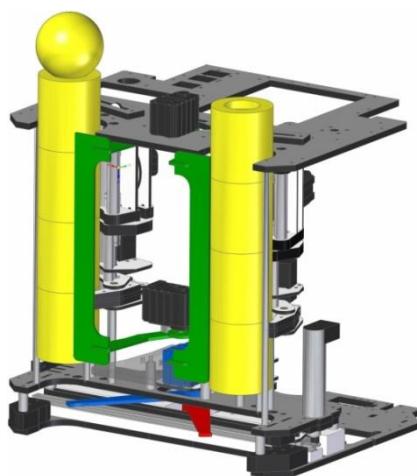


Figura 4-42 Secuencia foco3

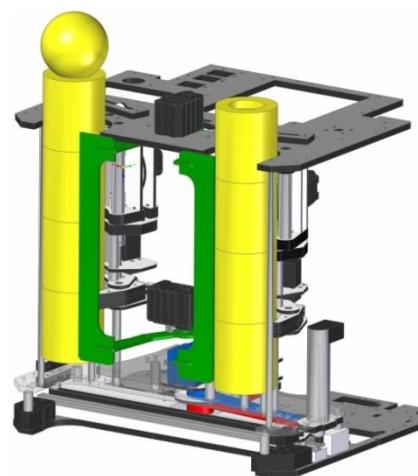


Figura 4-43 Secuencia foco 4

3. Las pinzas elevan los stands inferiores. En la columna izquierda es un paso necesario para la construcción de la torre y en la derecha para permitir que ambas paletas se oculten tras la columna derecha.
4. Ambas paletas son colocadas tras la columna derecha.

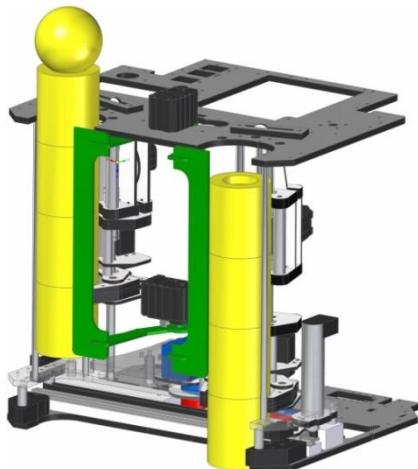


Figura 4-44 Secuencia foco 5

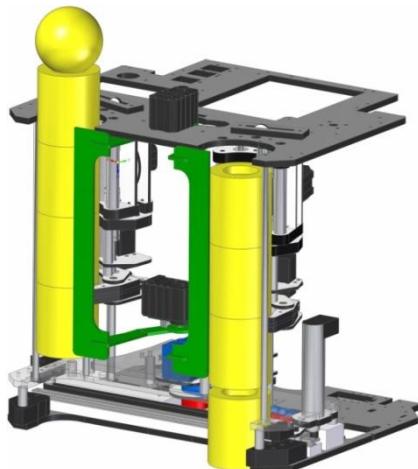


Figura 4-45 Secuencia foco 6

5. Se baja la pinza derecha y se abre.

6. Se eleva la pinza derecha hasta la altura de 70 mm se agarra el stand y se concluye la elevación hasta la cota máxima a 80 mm. Con este desplazamiento de 10 mm se libera el stand inferior del peso de la torre, para evitar problemas durante el inicio del desplazamiento

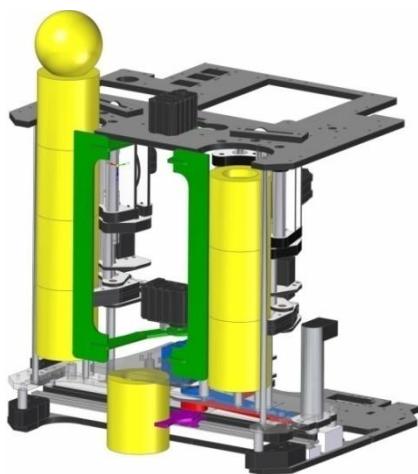


Figura 4-46 Secuencia foco 7

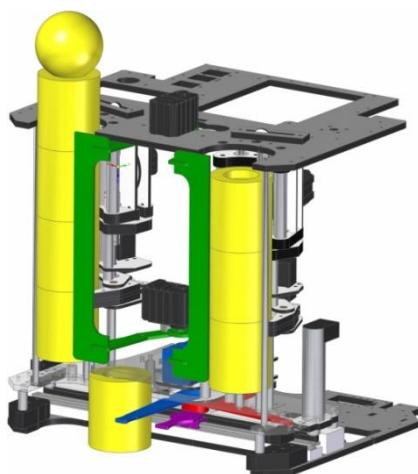


Figura 4-47 Secuencia foco 8

7. Se desplaza el stand por medio del carro horizontal (de color morado en la imagen) hasta la mitad del recorrido. Inmediatamente alcanzada esta posición inicia el retroceso hasta su posición inicial para prepararse para arrastrar otro stand.

8. Alcanzada esta posición la paleta superior (de color azul en la imagen) comienza el giro para completar el desplazamiento del stand hasta la posición de bloqueo izquierda.

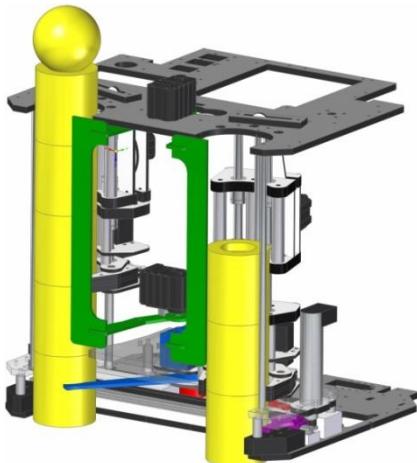


Figura 4-48 Secuencia foco 9

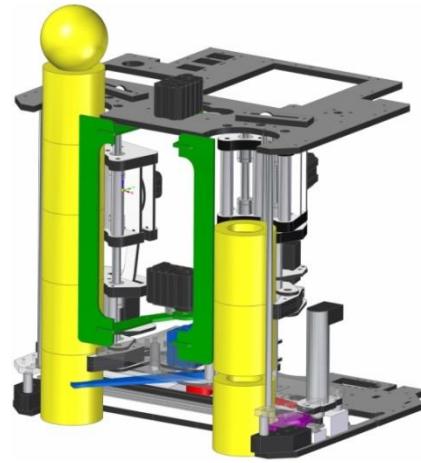


Figura 4-49 Secuencia foco 10

9. De forma paralela al bloqueo del stand por la paleta en el almacén izquierdo y una vez que el carro horizontal a alcanzado la posición inicial la pinza derecha desciende con el siguiente stand.

10. Inmediatamente concluido el bloqueo del stand izquierdo la pinza izquierda libera el stand inmediatamente superior dejándolo caer sobre el bloqueado y comienza el descenso hasta la posición inferior.

En el almacén derecho la pinza ha soltado el stand, se ha elevado hasta la altura de 70 mm, ha agarrado el stand inmediatamente superior y lo ha elevado hasta la cota máxima a 80 mm.



Figura 4-50 Secuencia foco 11



Figura 4-51 Secuencia foco 12

11. La pinza izquierda ha elevado el stand y la paleta superior (de color azul) se ha ocultado detrás de la torre izquierda. De forma paralela el carro horizontal desplaza el stand de la torre derecha hasta la mitad del recorrido. Inmediatamente alcanzada esta posición inicia el retroceso hasta su posición inicial para prepararse para arrastrar otro stand.

12. Alcanzada esta posición la paleta inferior (de color rojo en la imagen) comienza el giro para completar el desplazamiento del stand hasta la posición de bloqueo izquierda.

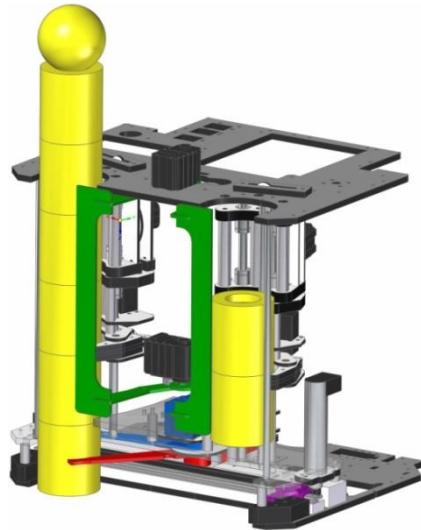


Figura 4-52 Secuencia foco 13

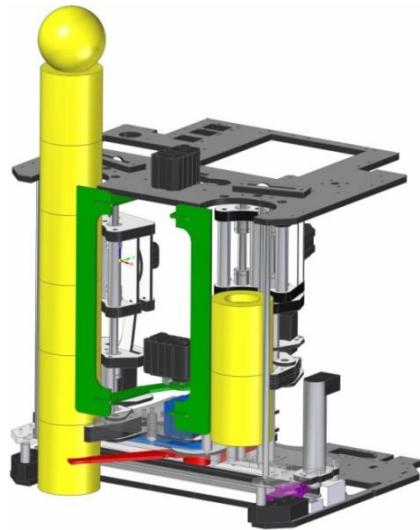


Figura 4-53 Secuencia foco 14

13. Bloqueado el stand por la paleta en el almacén izquierdo, comienza el descenso de la pinza izquierda. En el otro almacén se detienen las operaciones manteniendo el stand en la posición superior.

14. La pinza izquierda sujeta el stand bloqueado y comienza el ascenso.

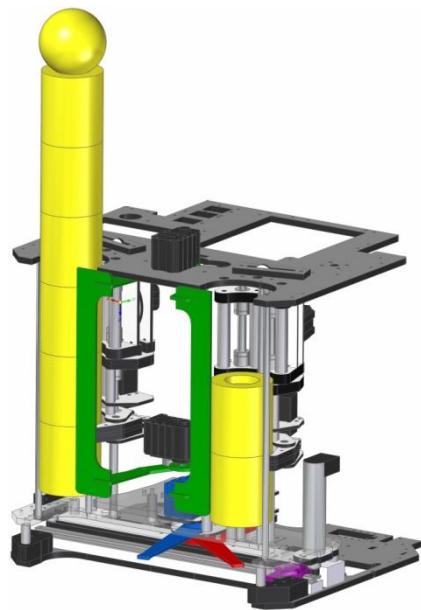


Figura 4-54 Secuencia foco 15

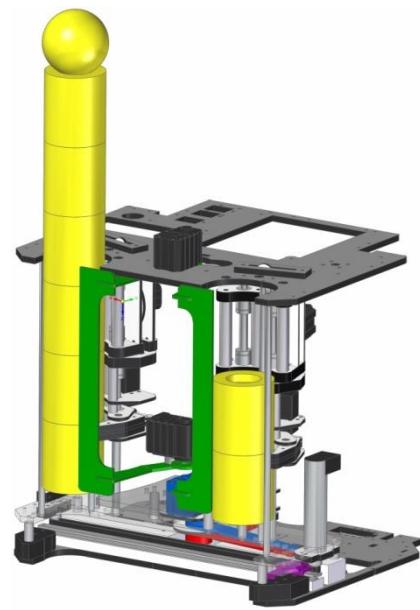


Figura 4-55 Secuencia foco 16

15. Una vez finalizado el ascenso de la pinza izquierda tanto la paleta inferior que bloqueaba el stand como la superior que permanecía oculta tras la torre izquierda comienzan a girar para colocarse tras la torre derecha.

16. Una vez que estas se encuentran tras la torre derecha el sistema se encuentra exactamente en la misma composición que en el punto número 4, repitiéndose desde este punto la misma secuencia hasta el 14.

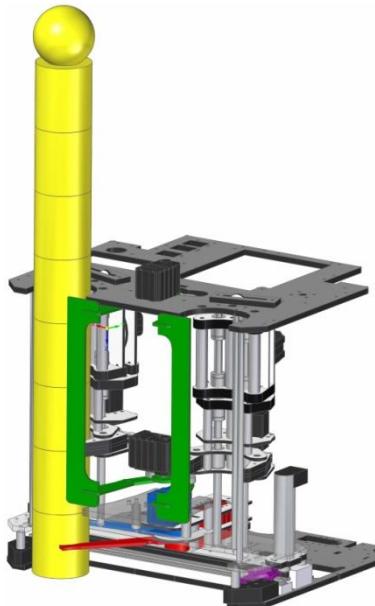


Figura 4-56 Secuencia foco 17

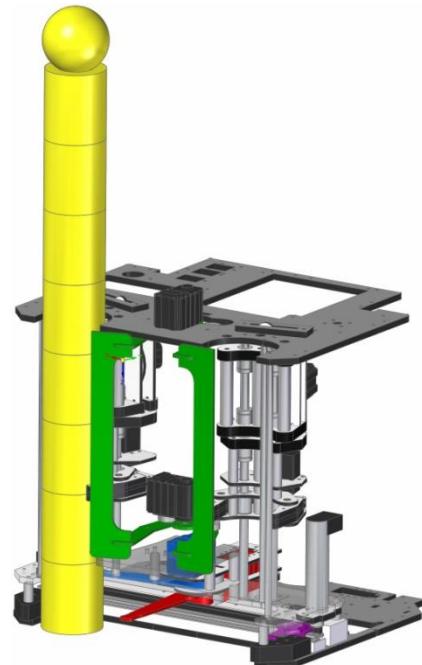


Figura 4-57 Secuencia foco 18

17. Esta secuencia es una repetición de la número 14, punto en el que se habrá concluido la construcción del foco a ocho niveles.

18. En esta secuencia la torre es liberada retirando la paleta que sujetaba el stand inferior, abriendo la pinza del carro izquierdo que agarraba el stand de la segunda altura y abriendo también la puerta que sujeta los stands de los niveles 2, 3, 4 y 5.

Una vez concluida la liberación el robot debe retroceder en línea recta a baja velocidad para evitar que la torre se pueda desestabilizar.

En la Figura 4-58 con una vista superior de la secuencia número 18 finalizada se muestran las posiciones de apertura máxima de la pinza (en color blanco en la imagen), de la puerta (en color verde) y de la paleta en la dirección de avance.

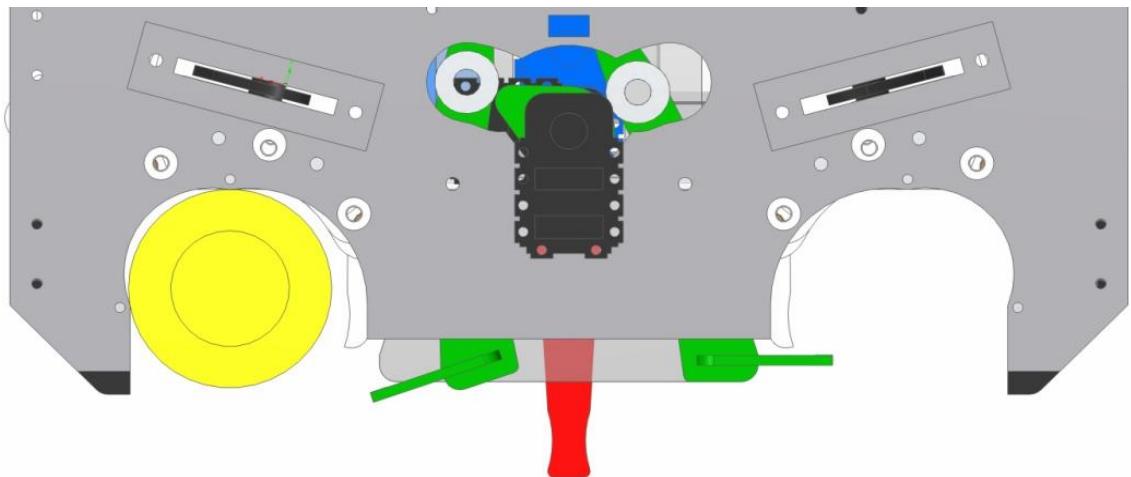


Figura 4-58 Detalle apertura máxima de puerta y pinza para liberación de foco

## 4.7. Tiempo mínimo de procesado de stands

En el apartado 4.1 "Prototipos y estudio previo" se calculó el mínimo tiempo de maniobra de procesado de un stand exigido en las especificaciones, resultando un tiempo de 1.72 segundos. Concluido el diseño de todos los sistemas se va a estimar el tiempo mínimo de maniobra conseguido.

Este tiempo se calculó como la suma del tiempo de desplazamiento y operación de otros sistemas entre las etapas 2 y 4 de la estrategia inicial. De forma paralela el sistema de manipulación de stands, en este caso funcionando como almacén tendrá que realizar las siguientes operaciones para quedar preparado para la recogida de otro stand en el mismo almacén. Estas serán:

1. Cierre de paleta hasta posición de bloqueo de stand, la paleta parte de la posición de recepción de stand a 90° (en la dirección de avance). En el apartado 4.2 "Sistema de paletas de stands" se calculó un tiempo de 0.103 segundos correspondiente al giro de 38° de un servo AX12 con una velocidad de 59 rpm.
2. Descenso de carro pinza desde posición superior de equilibrio hasta posición inferior de recogida de stand. Este tiempo nunca va a ser inferior al de la velocidad máxima del servo AX12 que para 180° corresponde a 0.49 seg.
3. Tiempo de cierre de pinza. Se ha calculado un tiempo de 0.17 seg.
4. Tiempo de ascenso de carro pinza hasta posición superior. Este tiempo no ha podido ser calculado con precisión, por lo que se va a utilizar para el cálculo un tiempo muy negativo un 25% superior a la velocidad máxima. Concretamente 0.61 seg.

No se ha tenido en cuenta el tiempo de apertura de la paleta hasta la posición de recepción (en dirección al avance) ya que se produce en paralelo con el de elevación que es muy superior a este.

$$t_{Almacenad\ oStand} = t_{BLOQUEO} + t_{BAJADA} + t_{AGARRE} + t_{SUBIDA} = \\ t_{Almacenad\ oStand} = 0.103 + 0.49 + 0.17 + 0.61 = 1.373\text{seg} < 1.72\text{ seg}$$

Por lo tanto se concluye que el sistema puede ser perfectamente capaz de cumplir las especificaciones requeridas, siendo el tiempo de procesado de stand menor del requerido.

Aunque en estos cálculos no se han tenido en cuenta ciertos tiempos muertos entre operaciones, se puede considerar que se van a compensar con los que tampoco fueron tenidos en cuenta para los movimientos entre las operaciones y el resto de operaciones intermedias.

## 5. Sistema de manipulación de palomitas

---

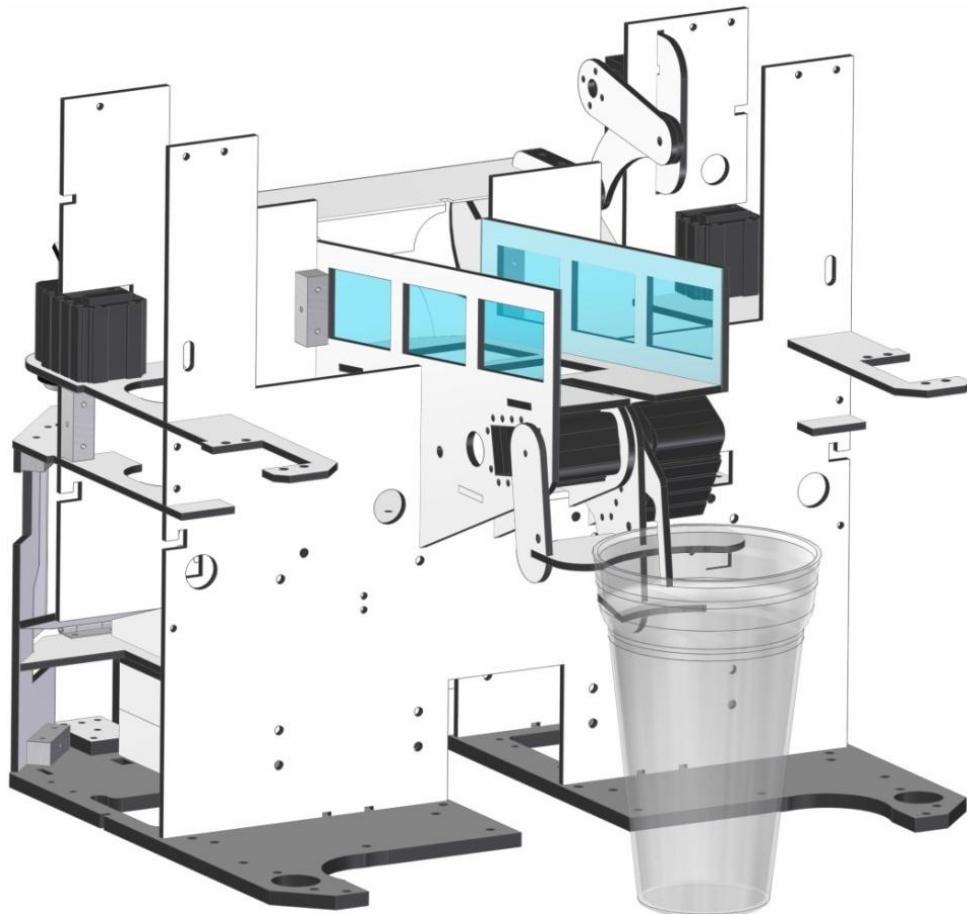


Figura 5-1 *Sistema de manipulación de palomitas con estructura portante Vista Delantera*

En este capítulo se recoge el diseño realizado para el sistema de manipulación de palomitas. En la 5.1 se muestra una imagen delantera de sistema integrado incluyendo las partes de la estructura portante sobre la que se sustentan los sistemas y en Figura 5-2 la vista trasera. Este sistema como se va a detallar en los próximos apartados está compuesto por tres subsistemas, que en este caso realizan tareas independientes y por lo tanto no van a requerir de una programación que sincronice sus movimientos. Los tres subsistemas son los siguientes:

- **SISTEMA MÁQUINAS DE PALOMITAS**

Este sistema está formado por tres subsistemas, dos de los cuales deben ser controlados de manera conjunta permitiendo vaciar las dos máquinas de palomitas de una forma rápida y sencilla. El tercero que permite vaciar las palomitas en el "cine" tiene la particularidad de que ha sido accionado utilizando el propio movimiento de las pinzas de captura del vaso trasero. Estos tres subsistemas con:

- **RAMPA DE PALOMITAS**

Se trata de un sistema doble simétrico formado por dos rampas giratorias accionadas por dos servos AX12 que introduciéndose por debajo de las pilas de palomitas las conducen de manera continua en la dirección de la rampa hacia el sistema de recogida que crea el siguiente subsistema.

- **BANDEJA DE PALOMITAS**

Es una bandeja pivotante que se abate ocupando el espacio entre ambos dispensadores de palomitas y que permite dirigir las palomitas empujadas lateralmente por las rampas hacia los depósitos en el interior del robot. Este sistema junto con el de rampas de palomitas debe ser controlado de manera coordinada para evitar que choquen entre sí.

Esta bandeja tiene una segunda función que consiste en impedir que cuando se encuentra en posición recogida, las palomitas depositadas en la rampa de descarga del vaso delantero hacia el trasera caigan en este.

- **PUERTAS DE PALOMITAS**

Se trata de un sencillo sistema de dos paredes que articuladas tienen la doble función de descargar las palomitas hasta las cestas existentes en los cines.

- **SISTEMA PINZA VASO TRASERO DE PALOMITAS**

Este sistema permite la captura de un vaso en la parte trasera del robot, mediante dos pinzas con forma circular accionadas por dos servos. El accionamiento de ambas debe de ser controlado de forma sincronizada para evitar que debido a su diseño puedan chocar. Los últimos grados de apertura de las pinzas han sido utilizados para accionar la apertura de las puertas de descarga de palomitas.

- **SISTEMA PINZA VASO DELANTERO DE PALOMITAS**

Este sistema permite la captura de un vaso en la parte delantera del robot, mediante un sistema de una pinza articulada sobre un brazo que por un lado permite ocultar el sistema para no superar el perímetro máximo antes del inicio del partido y por otro permite bascular las palomitas contenidas en el vaso a una rampa que atraviesa el robot y que las conduce hasta el interior del vaso capturado por el sistema de pinzas traseras. Como ya se ha explicado en el subsistema de la bandeja de las palomitas, está no permite la caída de las palomitas hasta el vaso trasero hasta que esta no se abre. Esta particularidad ha sido diseñada para evitar perder las palomitas basculadas por el vaso delantero si todavía no ha sido capturado el trasero.

Las especificaciones de este sistema se detallan dentro de diferentes secuencias de la estrategia inicial del apartado 2.1, las he agrupado en los siguientes puntos:

- Se deberá diseñar un sistema de recogida delantero en el eje de simetría del robot para la captura del vaso de la secuencia número 3 que deberá ser capaz de sujetarlo en un tiempo máximo de 0,2 segundos, que corresponde con el tiempo de giro de un servo Dynamixel AX12 para 60. Una vez recogido el robot deberá llevarlo consigo o almacenar las palomitas en su interior.

- El sistema de recogida del vaso trasero que deberá ser transportado durante el resto del partido para posteriormente dejarlo en la zona protegida de salida deberá tener las siguientes características:
  - Sistema con amplio margen de error de posicionamiento.
  - Sujeción robusta y preferiblemente alojada en el cuerpo del robot para evitar obstaculizar los giros y que otros robots pueden chocar si se encuentra en una posición prominente.
  - Se deberá prever un sistema para recuperar las palomitas del vaso de la posición 3 e introducirlas en este vaso.
- El sistema de recolección de palomitas deberá ser capaz de vaciar las palomitas de los dos dispensadores paralelos, por lo tanto será necesario diseñar un sistema doble simétrico. El diseño del robot en esta posición deberá permitir referenciar la posición de los encoder del sistema de odometría chocando con la pared o base de las máquinas dispensadoras. Para este sistema no se establece un límite de tiempo mínimo de operación, pero si se deberá asegurar un porcentaje de error mínimo y una alta tolerancia al error de posicionamiento en el eje X (longitudinal del campo).  
Se deberá diseñar un sistema capaz de vaciar las palomitas recolectadas en las máquinas de palomitas en las canastas de esta zona.

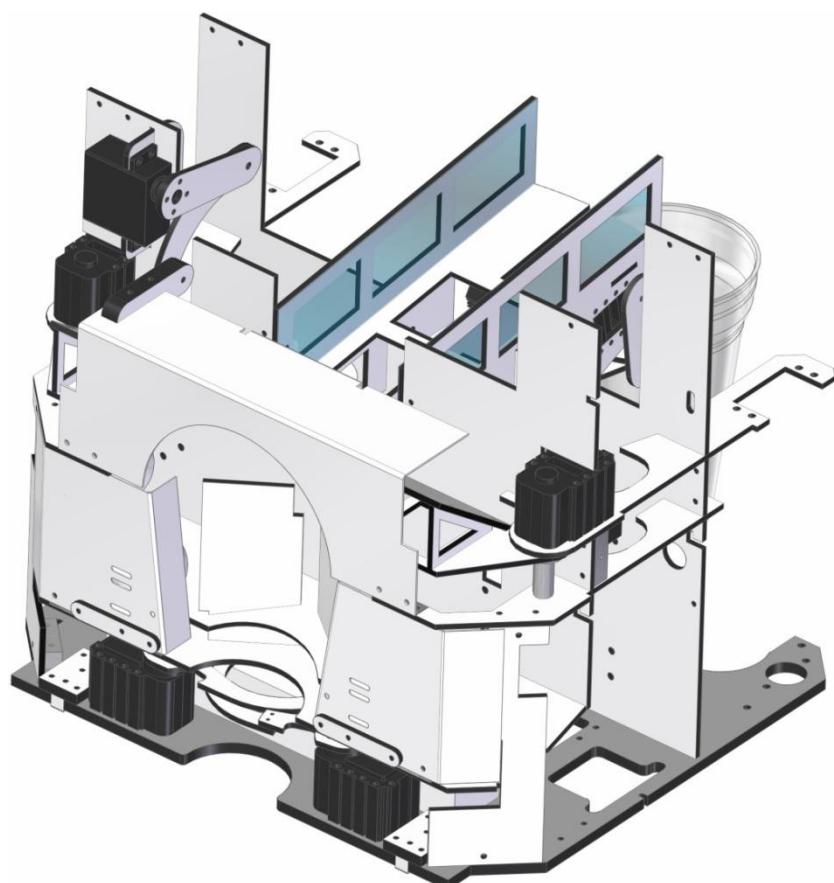


Figura 5-2 Sistema de manipulación de palomitas con estructura portante Vista Trasera

## **5.1. Prototipos y estudio previo**

---

Aunque el resultado final del diseño de estos sistemas han sido unos mecanismos de una concepción sencilla al igual que su implementación final que también ha resultado sencilla. La resolución del problema inicialmente planteado no resulta evidente ni única por supuesto. Partiendo siempre de la premisa fundamental de que el vaciado de las dos máquinas de palomitas contiguas debía realizarse de forma paralela para minimizar el tiempo total, se barajaron y estudiaron diferentes opciones, llegándose a construir incluso prototipos operativos rápidos de algunas de ellas.

Las dos opciones que fueron estudiadas en más profundidad fueron un sistema de succión a través de unas conducciones tubulares y un ventilador de gran potencia y el sistema finalmente implementado. Se han incluido videos e imágenes de pruebas de los prototipos creados en el directorio "*VIDEOS\PROTOTIPOS\MAQUINAS PALOMITAS*" de la documentación digital.

El sistema de succión fue descartado rápidamente al comprobar el alto grado de incertidumbre del éxito de la operación y principalmente por el gran espacio necesario para implementarlo. Frente a esta solución el sistema finalmente implementado resultaba más sencillo y mucho más compacto.

Para el accionamiento de los mecanismos diseñados se ha seguido la misma sistemática que para el sistema de manipulación de stands. Se ha primado la utilización de servos digitales AX12, debido a las ventajas que ofrecen para su control a través de una misma línea de datos y la lectura de la posición real lo que ha posibilitado la detección de bloqueos en los movimientos de los diferentes mecanismos. Por otro lado la utilización de estos servos con un elevado par motor para el accionamiento de unos mecanismos con unos requerimientos de par mínimos ha hecho innecesaria la realización de cálculos físicos, siendo los únicos necesarios los relativos a los tiempos de operación en función de las velocidades máximas de los servos utilizados.

## **5.2. Sistema máquinas de palomitas**

---

Para entender mejor las características de los tres subsistemas diseñados se van a explicar las secuencias de manipulación finalmente implementadas, a través de una secuencia de imágenes del diseño definitivo del robot simulando las operaciones dentro del campo de juego.

### *DESPLEGADO DE SUBSISTEMAS DE RECOLECCIÓN*

Los dos subsistemas deben de ser desplegados antes de avanzar hasta la posición de recolección. En primer lugar se debe abrir la bandeja y a continuación las rampas. Es necesario seguir este orden debido a que la bandeja se cierra sobre las rampas. Se puede ver la posición del robot y el estado de los subsistemas en la Figura 5-3 y Figura 5-4.

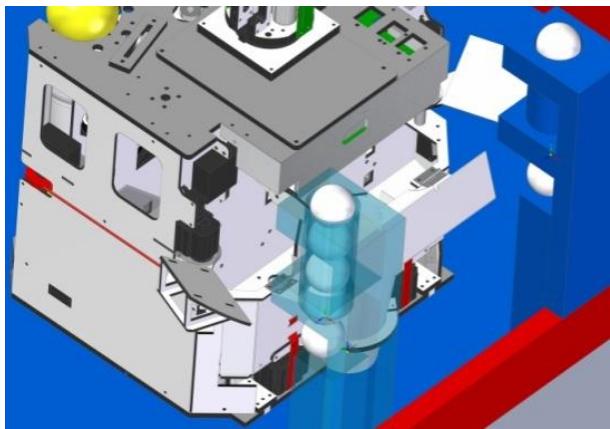


Figura 5-3 Sistema máquina de palomitas estado previo a entrada a máquinas de palomitas en campo Vista Perspectiva

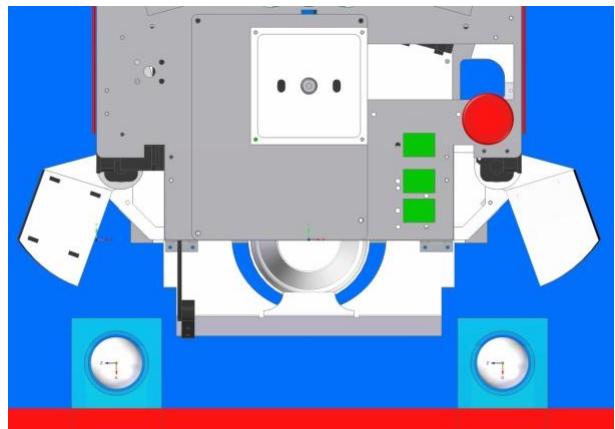


Figura 5-4 Sistema máquina de palomitas estado previo a entrada a máquinas de palomitas en campo

### *MOVIMIENTO DE RETROCESO HASTA POSICIÓN DE RECOLECCIÓN*

El ángulo de apertura de las rampas debe permitir el avance del robot hasta apoyarse contra la base de las máquinas sin tocar en este desplazamiento la primera palomita. Este punto de apoyo será utilizado para el referenciado del sistema odométrico en el eje Y del campo. En la Figura 5-5 y Figura 5-6 se muestra la posición previa a la recolección.

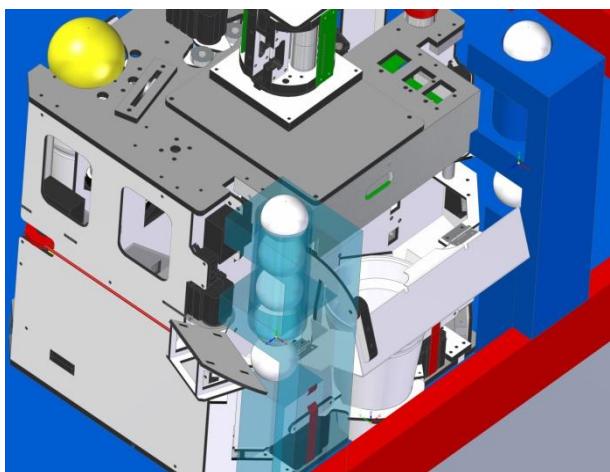


Figura 5-5 Sistema de palomitas posición y estado previo a recolección de palomitas en campo Vista en perspectiva

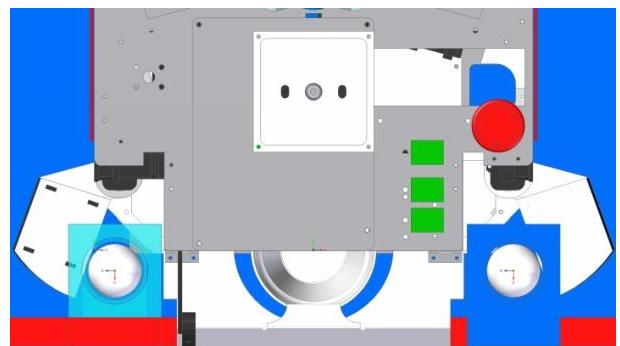


Figura 5-6 Sistema de palomitas posición y estado previo a recolección de palomitas en campo

### *RECOLECCIÓN DE PALOMITAS*

La operación de recolección se realiza girando las rampas hasta introducirlas bajo el dispensador vertical de palomitas. La primera es empujada y el resto dirigidas por medio de las rampas a la bandeja que las recibe e introduce en el interior de los depósitos. En la Figura 5-7 y Figura 5-8 se muestra la posición de las rampas y la simulación de la recolección. Las máquinas de las palomitas aparecen semi-transparentes u ocultas para facilitar la visión.

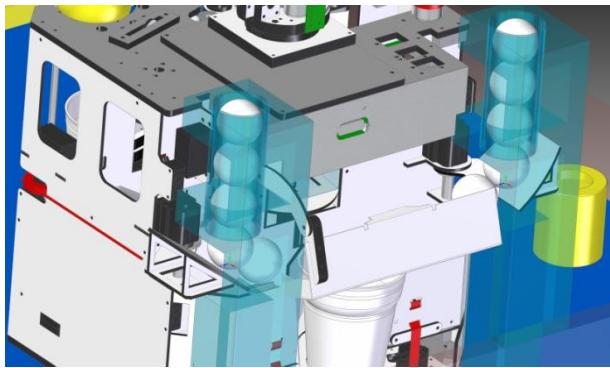


Figura 5-7 Sistema de palomitas con rampas recolectando palomitas en campo Vista Perspectiva

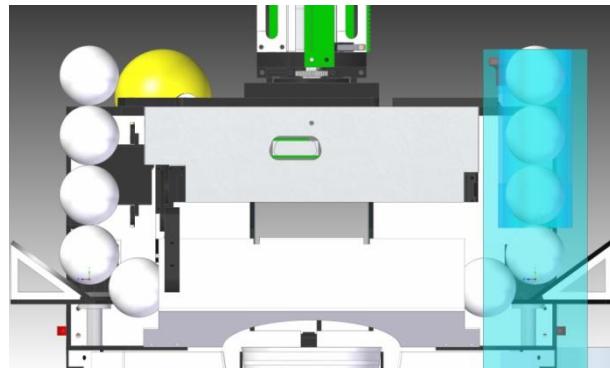


Figura 5-8 Sistema de palomitas con rampas recolectando palomitas en campo Vista Frontal

Concluido el tiempo estimado de recolección, que ha sido definido tras la realización de pruebas de funcionamiento, los dos subsistemas pueden ser replegados desde la misma posición de recolección.

#### VACIADO DE PALOMITAS EN CESTAS DE CINE

Alcanzada la posición exacta de vaciado, las puertas se abren por el accionamiento indirecto del movimiento de apertura de las pinzas de sujeción del vaso trasero, en su ángulo de apertura máximo. Ambas puertas dirigen las palomitas hasta las cestas guiadas por sus pliegues laterales. En la Figura 5-9 y Figura 5-10 se muestra la posición exacta de descarga del robot y el ángulo de apertura de las puertas. El lateral del campo aparece en color rojo semitransparente para facilitar la visión de las cestas.



Figura 5-9 Apertura puertas palomitas en campo Vista Lateral

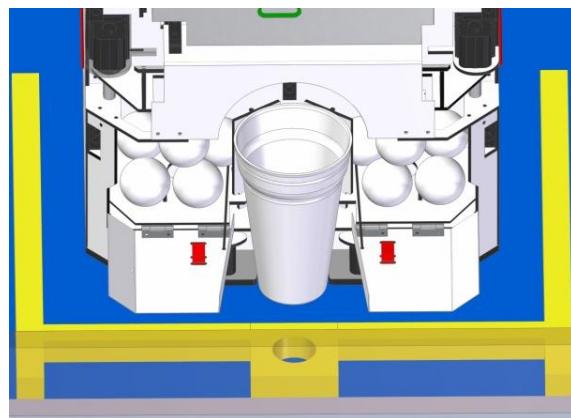


Figura 5-10 Apertura puertas palomitas en campo Vista Frontal

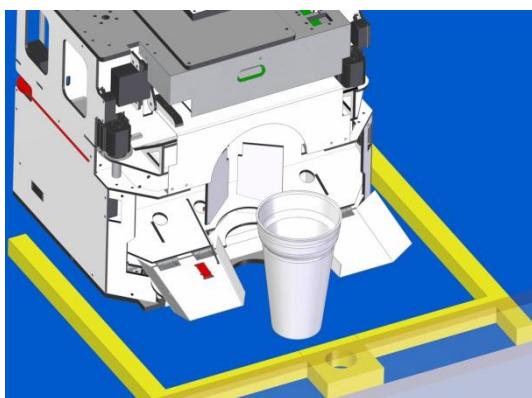
***LIBERACION DE VASO TRASERO Y PLEGADO DE PINZAS***

Figura 5-11 Avance de robot para liberación de vaso en campo

Movimiento de avance del robot manteniendo las pinzas en posición de apertura máxima.

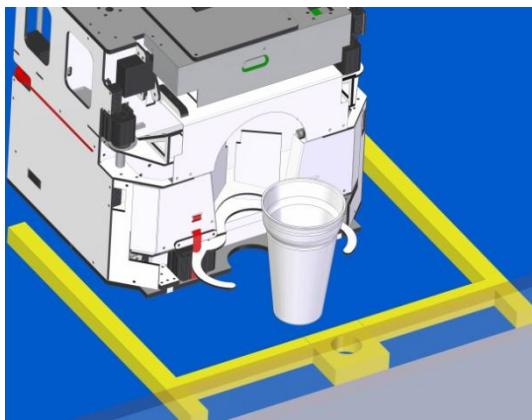


Figura 5-12 Cierre de puertas de palomitas en campo

Cierre de pinzas hasta posición previa a captura de vaso. En esta posición las puertas se elevan automáticamente por un sistema que será explicado más adelante.

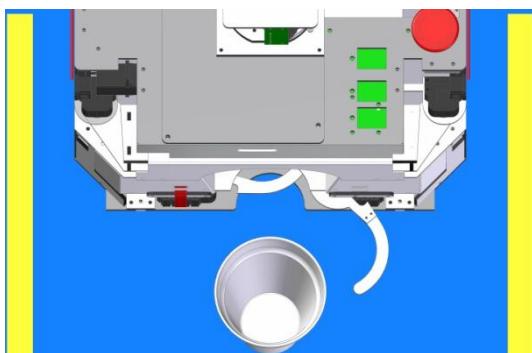


Figura 5-13 Cierre de pinza izquierda en campo

Cierre de pinza izquierda.

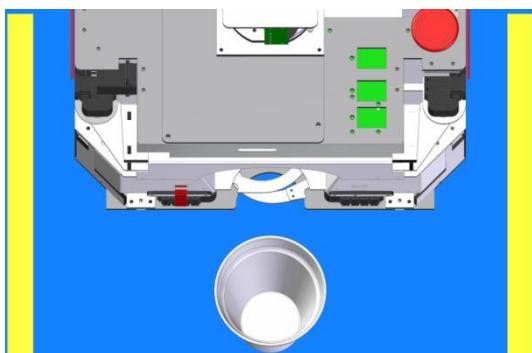


Figura 5-14 Cierre de pinza derecha en campo

Cierre de pinza derecha sobre pinza izquierda. Como se va a explicar el diseño realizado obliga al movimiento coordinado de las pinzas para evitar el choque de ambas.

### 5.2.1. Rampas de palomitas

Para el diseño de las rampas se realizaron una serie de prototipos rápidos, para analizar el comportamiento de salida de las pelotas. El diseño de estas tambien se solapo con el de la morfología de la planta del robot ya que era preciso poder crear un sistema simétrico que recogiese de forma paralela las dos máquinas contiguas. Este punto fue desarrollado en el apartado 2.3 "Analisis y dimensionado de sistemas" y definió el ancho final del robot.

En la siguiente secuencia de imágenes se muestran las tres posiciones de giro definidas para la maniobra de recolección.

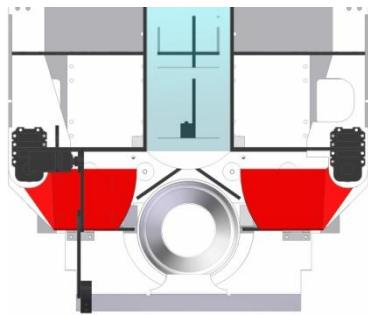


Figura 5-15 Rampas en posición recogida Vista Superior

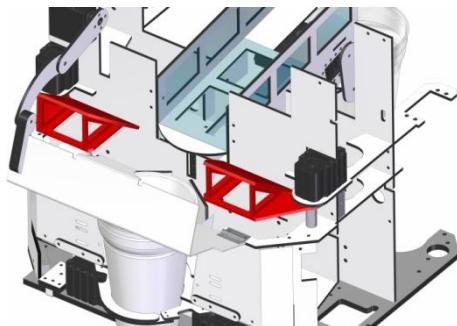


Figura 5-16 Rampas en posición recogida Vista Perspectiva

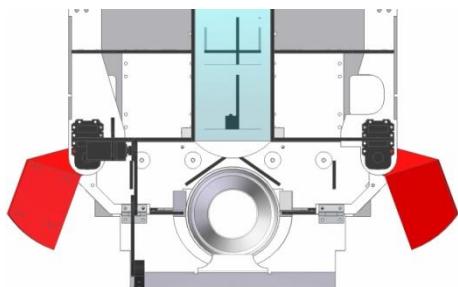


Figura 5-17 Rampas en posición de entrada a recolección Vista Superior

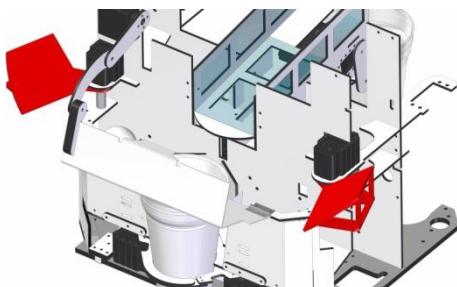


Figura 5-18 Rampas en posición de entrada a recolección Vista Perspectiva

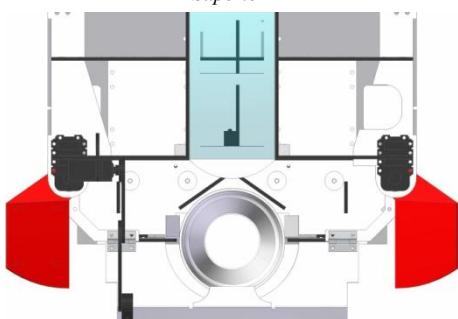


Figura 5-19 Rampas en posición de recolección Vista Superior

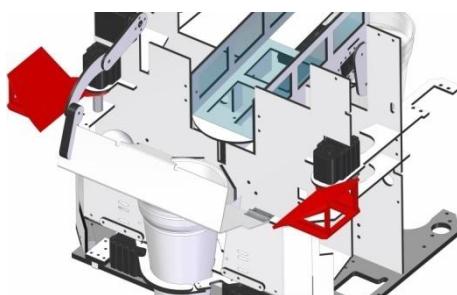


Figura 5-20 Rampas en posición de recolección Vista Perspectiva

En la Figura 5-15 y Figura 5-16 se muestra en diferentes vistas la posición de las rampas recogidas. En esta posición se colocan como tapa de los depósitos de las palomitas.

En la Figura 5-17 y Figura 5-18 se muestra la posición de las rampas con la apertura necesaria para la entrada a la posición de recolección sin tocar las palomitas.

En la Figura 5-19 y Figura 5-20 se muestra la posición de las rampas en modo de recolección. Tras diferentes pruebas se ha comprobado que no es necesario agitar las rampas durante la recolección para asegurar la caída de todas las palomitas, habiendo tenido una efectividad del 100%.

Ambas rampas están accionadas por dos servos AX12 que han permitido controlar la posición alcanzada por los servos y detectar posibles bloqueos durante el giro con pelotas en el interior de los depósitos. La transmisión del movimiento ha sido directa desde el eje de salida del servo.

El diseño realizado ha creado un sistema de giro robusto, mediante la utilización un eje de suplemento denominado "Suplemento Servo Brazos Palomitas", creando un conjunto de giro articulado sobre dos apoyos, la propia cabeza del servo y otro en el extremo del suplemento incorporado. Con este sistema se han reducido los esfuerzos por brazo de palanca sobre la cabeza del servo. Ver la Figura 5-21.

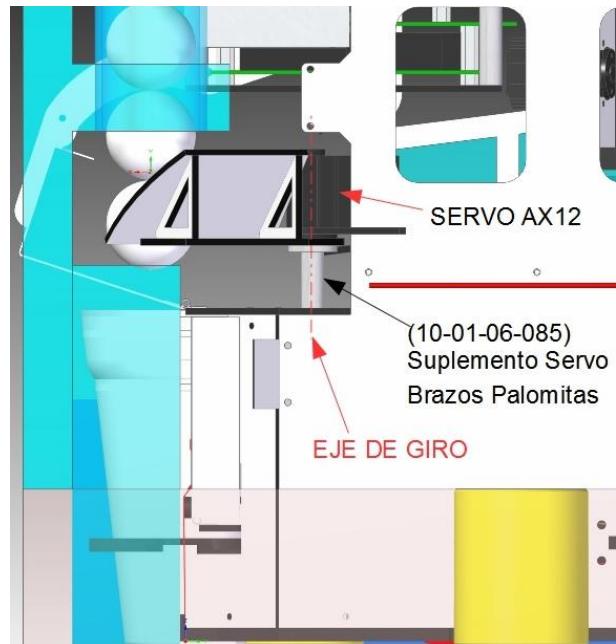


Figura 5-21 Sistema de palomitas posición y estado previo a recolección de palomitas en campo Vista lateral

### 5.2.2. Bandeja de palomitas

El diseño de esta bandeja en el conjunto de la parte trasera del robot, ha obligado a desarrollar un sistema de accionamiento a través de una biela articulada mediante un eje en la parte superior de la bandeja y el otro en el extremo de una manivela pivotante a través de un servo. Este sistema ha permitido dejar libre todo el espacio inferior de la bandeja, para permitir que las palomitas no encuentren obstáculos en su caída hacia los depósitos.

Este sistema con cuatro ejes articulados ha sido diseñado mediante sucesivas aproximaciones en dibujo en dos dimensiones, hasta conseguir el movimiento angular deseado con un giro determinado en el eje del servo. En la sucesión de figuras Figura 5-22, Figura 5-23, Figura 5-24 y Figura 5-25 se muestra una recreación del movimiento desde la posición de repliegue hasta la de máxima apertura o recolección. El software de diseño 3D utilizado ha permitido la simulación de todo este tipo de movimientos, permitiendo depurar los diseños realizados previamente en 2D analizando las interferencias entre las distintas piezas durante los movimientos.

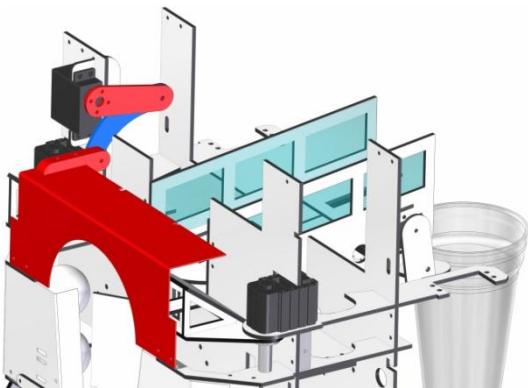


Figura 5-22 Sistema bandeja posición recogida

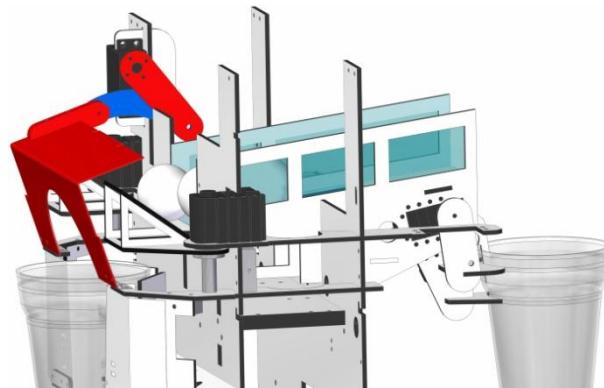


Figura 5-23 Sistema bandeja posición apertura 1

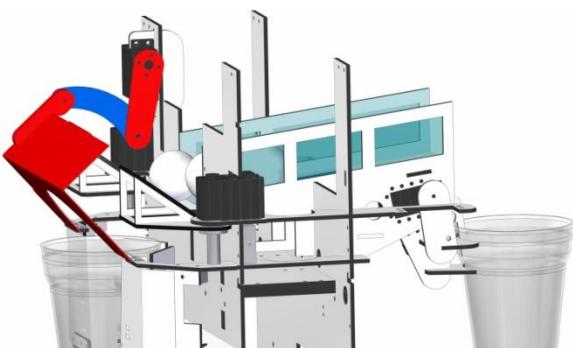


Figura 5-24 Sistema bandeja posición apertura 2

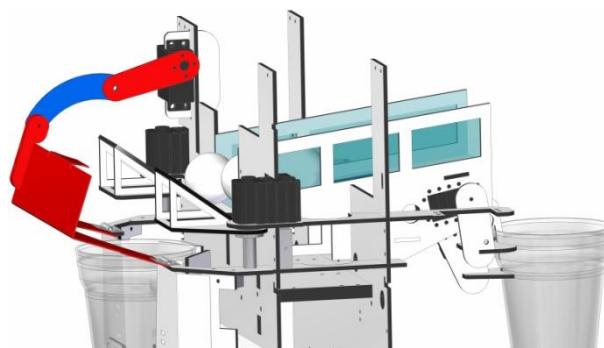


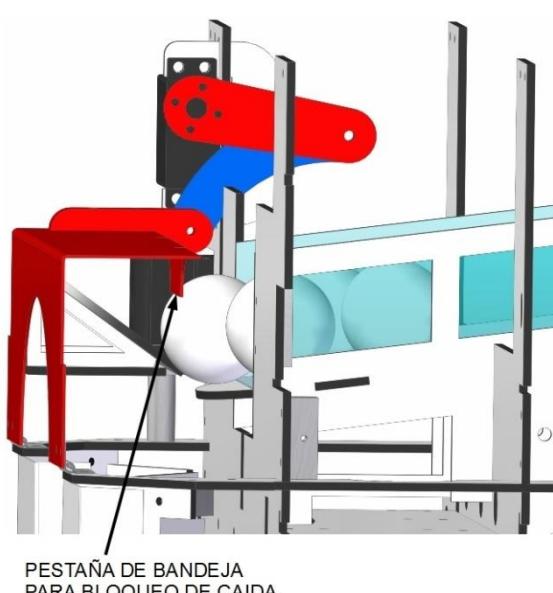
Figura 5-25 Sistema bandeja posición apertura de recolección

Se han coloreado de distinto color las piezas del conjunto para facilitar la comprensión de su funcionamiento. La pieza roja superior corresponde con la manivela unida a la cabeza del servo, esta está unida a la biela que aparece en color azul y por último el resto de la bandeja se ha coloreado también de color rojo.

El vaciado de media circunferencia practicado en la bandeja tiene como misión permitir que el vaso trasero pueda contener palomitas por encima de su borde superior. Esta situación es probable debido a que el sistema de bandeja no evita que puedan caer palomitas de las máquinas en el interior de este, que con las cuatro que contiene inicialmente más las cuatro del vaso delantero ya se encuentra cerca de este límite.

Por último ha sido necesario incluir una pestaña en la bandeja, para que encontrándose replegada evite la caída de las palomitas de la rampa del vaso delantero en el trasero. Estas solo podrán caer cuando se abra al menos en parte la bandeja. En la Figura 5-26 se muestra la situación y el funcionamiento de esta pestaña.

En la Figura 4-28 se muestra una vista superior del sistema con una simulación de la caída de las palomitas, una vez abierta la bandeja.



PESTAÑA DE BANDEJA  
PARA BLOQUEO DE CAÍDA  
DE PALOMITAS DE RAMPA

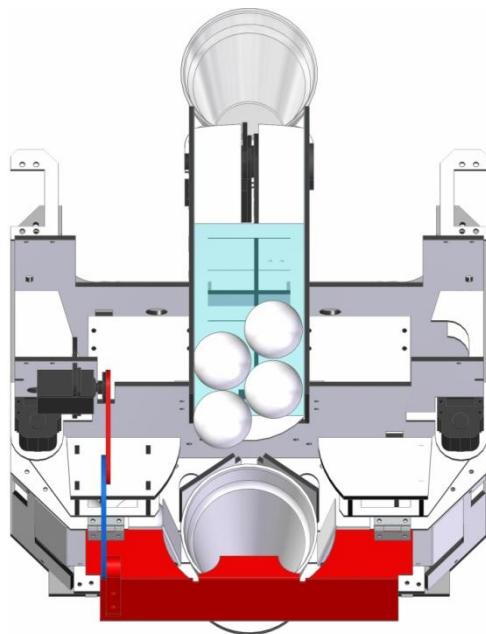


Figura 5-27 Detalle caída palomitas en vaso trasero tras apertura de bandeja

### 5.2.3. Puertas de palomitas

En el desarrollo de este sistema la mayor dificultad planteada ha sido el espacio casi inexistente para la ubicación del sistema de accionamiento. Finalmente el sistema diseñado ha sido el menos mecánico desde el punto de vista de una solución industrial digamos, pero probablemente ha sido el más ingenioso, el que menos recursos a necesitado y uno de los más eficientes, teniendo como algunos otros una efectividad durante la competición del 100%, no habiéndose producido ningún fallo de funcionamiento.

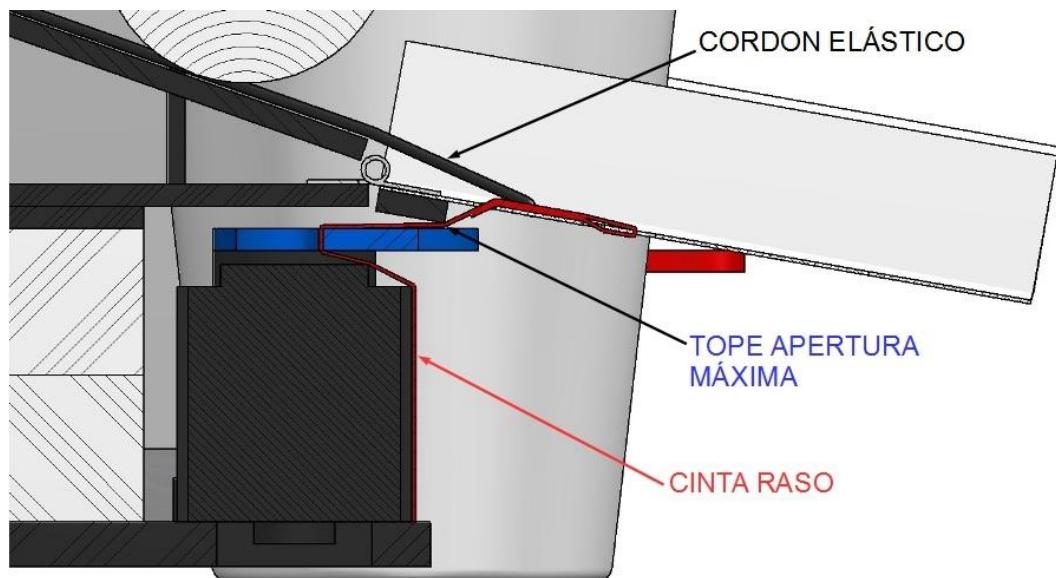


Figura 5-28 Vista en sección de sistema de apertura de puerta de palomitas con detalle de cinta de tracción para apertura con giro de pinza de vaso trasero

En la Figura 5-28 se presenta una vista en sección perpendicular al eje de articulación de las puertas de las palomitas, en la que se pueden observar varios detalles:

- El sistema de accionamiento del cierre de las puertas, que evita que estas puedan abrirse, bajo el peso de las palomitas que es mínimo, es un cordón elástico con un extremo unido a la puerta y el otro al interior de la estructura del depósito. Mediante la colocación de un tope de máxima apertura de la puerta, se ha asegurado que este cordón elástico siempre mantiene un momento de fuerza sobre el eje de articulación (en función de la distancia mínima del eje a la dirección de la fuerza) suficiente para garantizar que desde esta posición de máxima apertura y mínimo momento, va a ser capaz de comenzar la elevación de esta.
- La apertura se realiza a través de una cinta de "raso" de 10mm de ancho (en color rojo en la imagen), que con las puertas cerradas permanece tensada en dirección vertical y con la apertura de las pinzas a partir de un determinado ángulo es forzada a introducirse hacia el interior de la estructura del robot, produciendo la apertura de la puerta. Para asegurar un momento mínimo, al igual que en el caso anterior, se ha colocado un suplemento que también cumple las funciones de tope de apertura y que aparece en la imagen pegado en color negro a la parte inferior de la puerta.

En el diseño de las pinzas se han incluido unos vaciados en su cara externa para asegurar que las cintas no tienden a desplazarse hacia los laterales del robot debido a la forma circular de estas. En la Figura 5-29 se puede apreciar este detalle, en esta imagen las pinzas aparecen en el ángulo previo al punto de contacto con las cintas.

En la Figura 5-30 se muestra el detalle de la apertura de la puerta con esta en estado semi-transparente para facilitar la visión del conjunto.

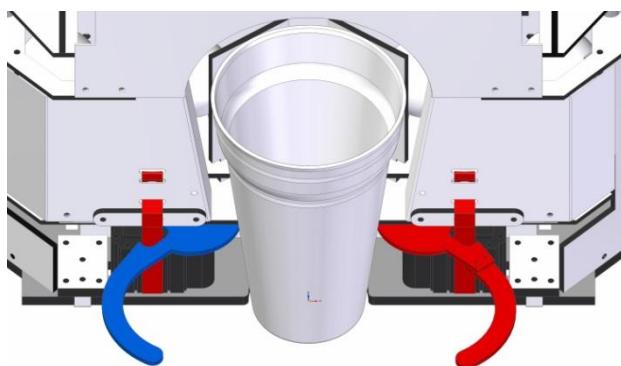


Figura 5-29 Angulo de pinzas de comienzo de apertura de puertas

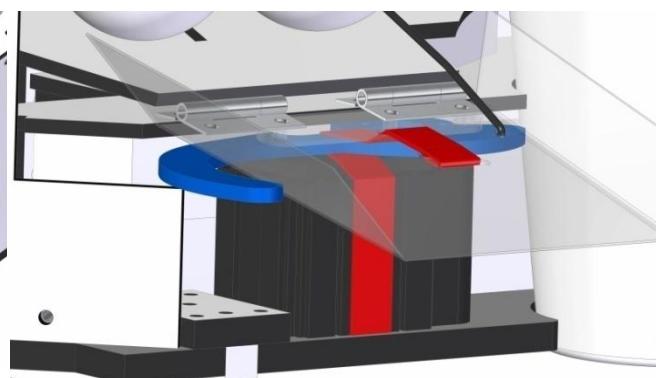


Figura 5-30 Detalle apertura de puerta de palomitas con puerta semi-transparente

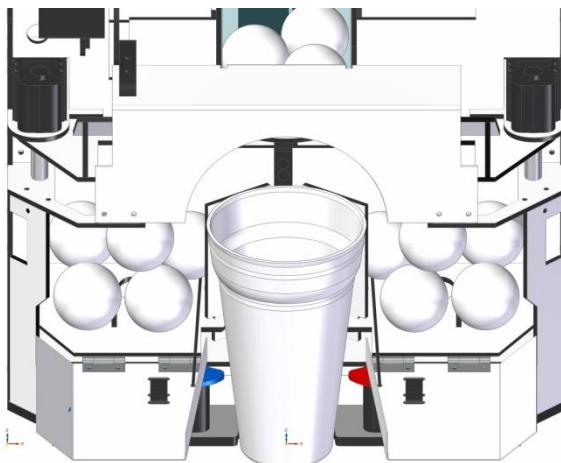


Figura 5-31 Vista Frontal de puertas de palomitas abiertas



Figura 5-32 Vista Lateral de puertas de palomitas abiertas

Otra parte importante del diseño ha sido garantizar la salida de todas las palomitas y en el menor tiempo posible. El sistema de puerta-rampa ha permitido liberar el contenido retirando toda la pared contenedora. Además estas puertas aprovechando la forma achaflanada de las esquinas del robot presentan unas pestañas que evitan la caída lateral. Ver Figura 5-31 y Figura 5-32.

### 5.3. Pinza vaso trasero de palomitas

La principal especificación para el diseño de este sistema ha consistido en asegurar un amplio margen de captura del vaso. Para conseguir este objetivo se diseñó un sencillo sistema con pinzas articuladas con dos servos independientes AX12 que garantizaban un margen de captura en la dirección de avance hacia el stand mayor de 60 mm, y lateralmente de 160 mm, que corresponden con la distancia de apertura de las pinzas (Ver Figura 5-33). También se ha incluido en el sistema un sensor E3Z-D61 de OMRON para la detección de que el vaso se encuentra dentro del alcance de captura.

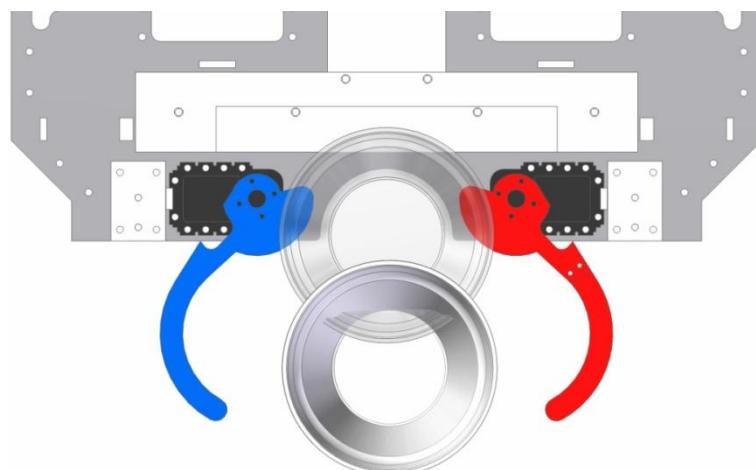


Figura 5-33 Margen de captura de vaso con pinzas traseras

Por problemas de espacio el sistema diseñado ha obligado a controlar las pinzas con una secuencia concreta para evitar el choque entre ellas durante la apertura y replegado de estas. Como se puede ver en la Figura 5-34 parte de la pinza derecha (en color rojo) se encuentra en el mismo plano horizontal que la izquierda. Por lo tanto es preciso iniciar la apertura de la derecha antes de comenzar la de la izquierda, ver Figura 5-35. En la operación de replegado el procedimiento es el inverso.

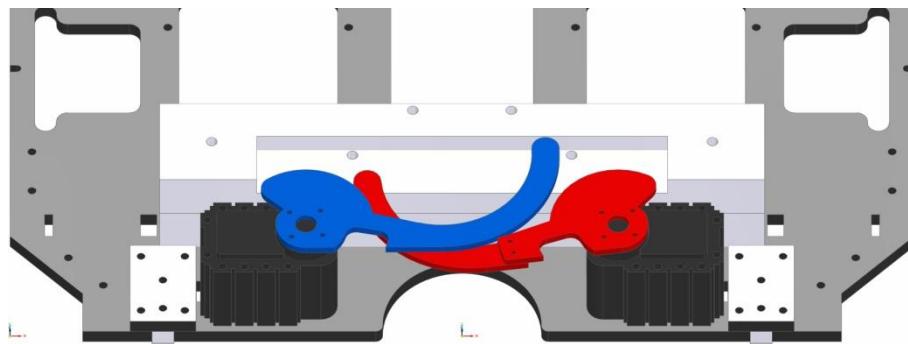


Figura 5-34 Detalle pinza vaso trasero cerradas

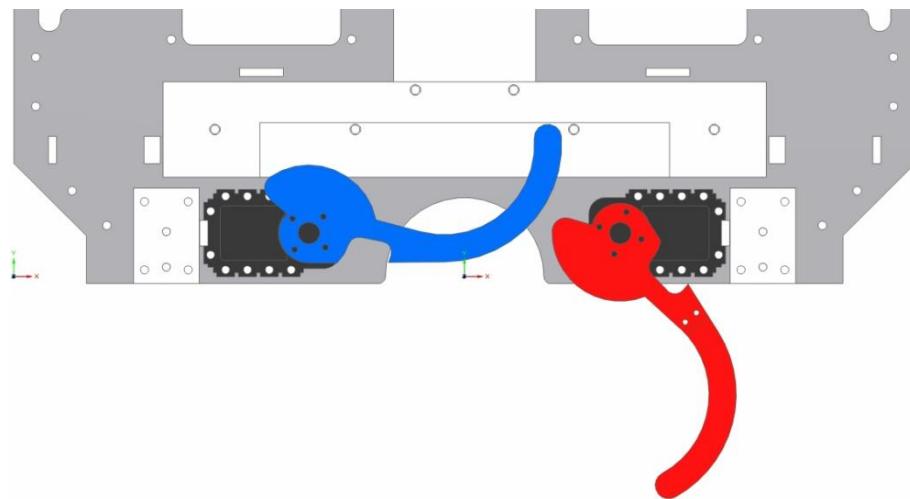


Figura 5-35 Detalle orden de cierre de pinzas de vaso trasero

Otra característica destacable exigida en las especificaciones era asegurar un bloqueo robusto del vaso para evitar su posible choque con otros robots. En la Figura 5-36 se puede observar que el sistema implementado introduce la mitad del vaso en el perímetro del robot, alojándolo contra tres semicircunferencias a diferente altura que estabilizan completamente su posición.

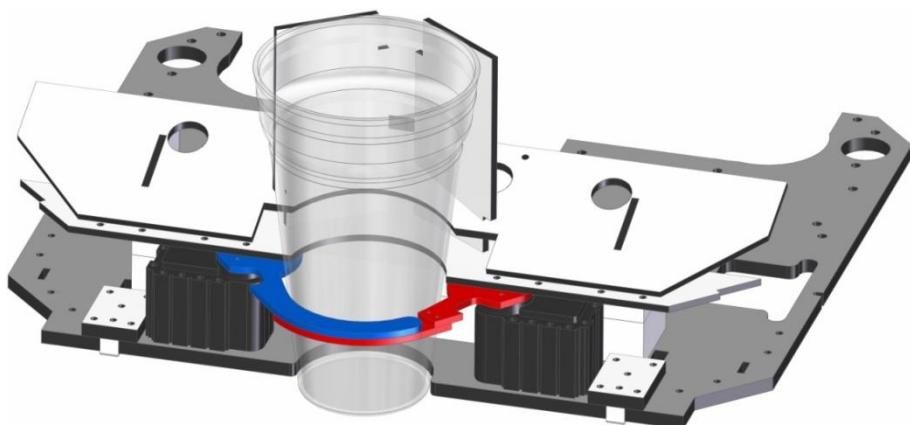


Figura 5-36 Detalle apoyo interno de vaso trasero

#### 5.4. Sistema pinza vaso delantero de palomitas

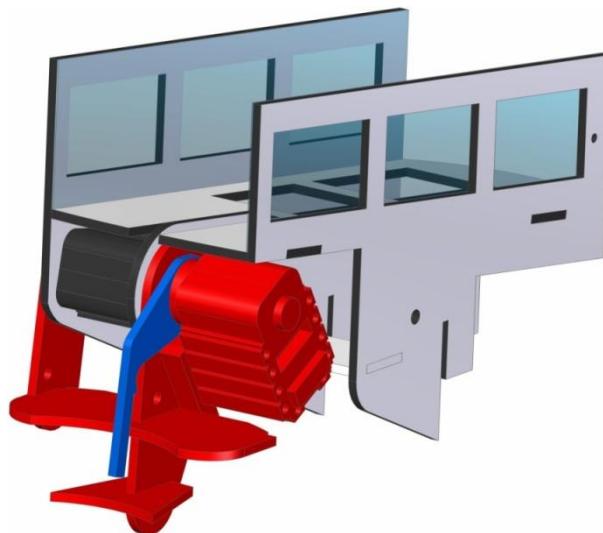


Figura 5-37 Sistema vaso delantero con partes móviles identificadas por colores

En la Figura 5-37 se muestran todas las piezas que conforman el diseño definitivo de este sistema. Los conjuntos articulados en esta figura y en las siguientes han sido coloreados en diferentes tonos para identificar fácilmente las partes móviles.

Como se observa se han creado dos partes articuladas a través de dos servos AX12. La pieza azul corresponde a la pinza de captura del vaso y esta accionada por un servo fijado en el otro conjunto articulado, denominado brazo y coloreado de color rojo. Este conjunto es accionado por un segundo servo solidario a la estructura del robot, y permite volcar el contenido del vaso capturado a la rampa de descarga hacia el vaso trasero. El diseño del

sistema pinza-brazo con tres puntos de apoyo para el bloqueo del vaso y siendo dos de ellos, semicircunferencias adaptadas al diámetro de este, han creado un sistema que además de autocentrar el vaso, imposibilita el desplazamiento del este en cualquier dirección.

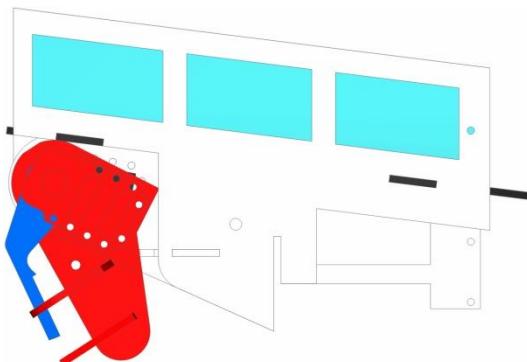
Para garantizar la estabilidad y robustez del brazo, se ha diseñado una estructura articulada entre dos ejes de giro lo suficientemente distanciados. Estos ejes son el propio plazo del servo en el lado derecho de la imagen de la Figura 5-37 y el eje auxiliar que disponen los propios servos AX12 al lado contrario del plato, este correspondería al lado izquierdo de la estructura de la imagen.

Constructivamente el conjunto del brazo tiene la característica de haber sido ensamblado por el sistema de pestañas y no se ha necesitado ningún tipo de fijación atornillada ni adhesiva.

La estructura de soporte del sistema ha sido aligerada con una serie de aberturas que posteriormente fueron cubiertas con acetatos transparentes, tal y como aparece en las imágenes.

Para comprender las características del diseño se va a explicar las diferentes posiciones de operación durante una secuencia de captura de un vaso.

#### *POSICIÓN REPLEGADA*



La articulación del brazo ha sido diseñada para permitir ocultar todo el sistema de captura por dentro del perímetro del robot, para no sobrepasar el máximo permitido antes del inicio del partido. En la Figura 5-38 se muestra la vista lateral derecha del sistema completamente replegado.

Figura 5-38 Sistema vaso delantero replegado Vista Lateral

#### *POSICIÓN PREVIA A CAPTURA*

Una de las especificaciones del diseño de este sistema ha sido el tiempo mínimo de captura, estimado inicialmente en 0,2 segundos, que corresponde con el tiempo de giro de un servo Dynamixel AX12 para 60°. Para reducir este tiempo se ha definido una posición intermedia en la que los servos se encuentran en una posición desplegada próxima a la de captura. De esta forma se reduce el ángulo a recorrer para completar la operación de bloqueo de vaso. En la Figura 5-39 y Figura 5-40 se muestra esta posición desde diferentes perspectivas.

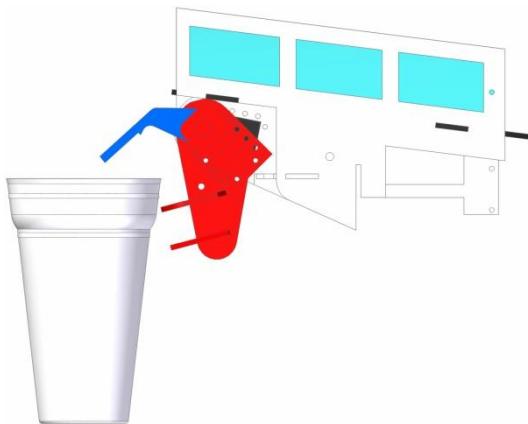


Figura 5-39 Sistema vaso delantero posición previa a captura  
Vista Lateral

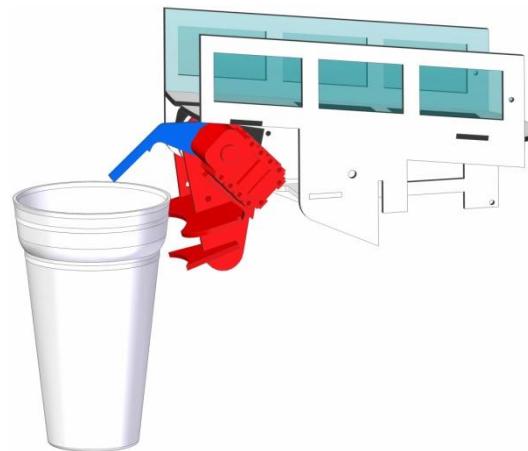


Figura 5-40 Sistema vaso delantero posición previa a captura  
Vista Perspectiva

Para la captura del vaso, una vez alcanzada la posición destino del robot, se ha ideado un sistema que evita el volcado accidental del mismo al ser empujado por un error de posicionamiento. El sistema consiste en un cierre simultáneo de brazo y pinza hasta la posición de captura, evitando avanzar hasta esta con el brazo en la posición de captura final. En la Figura 5-41 y Figura 5-42 se muestra la posición de captura desde diferentes perspectivas.

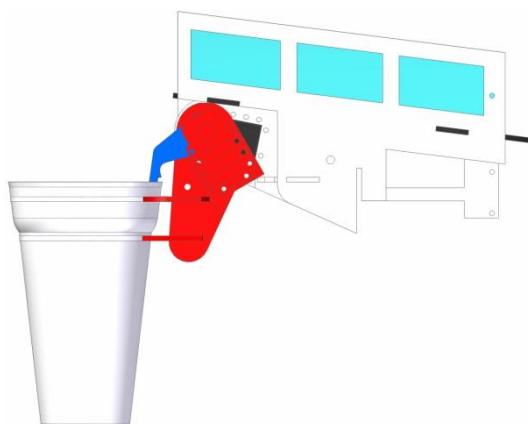


Figura 5-41 Sistema vaso delantero posición de captura Vista  
Lateral

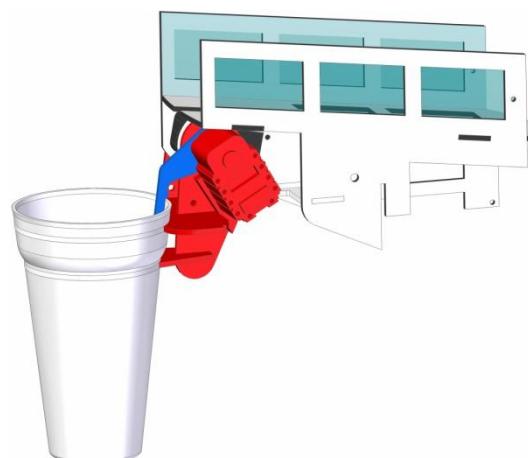


Figura 5-42 Sistema vaso delantero posición de captura Vista  
Perspectiva

Por último la maniobra concluye con el giro del brazo hasta la posición de descarga en la rampa del vaso trasero. En la Figura 5-43 y Figura 5-44 se muestra esta posición desde diferentes perspectivas.

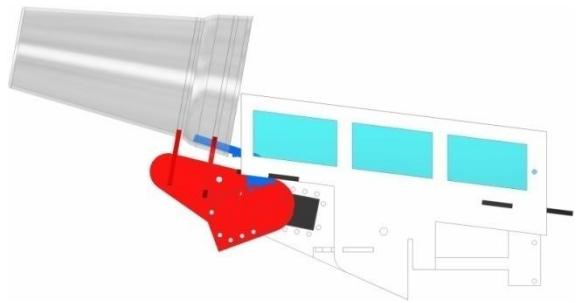


Figura 5-43 Sistema vaso delantero posición de vaciado Vista Lateral

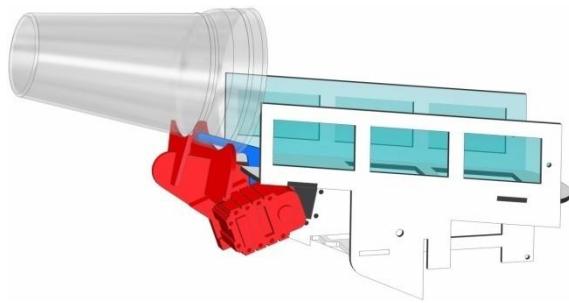


Figura 5-44 Sistema vaso delantero posición de vaciado Vista Perspectiva

## 6. Sistema de cierre de claquetas

---

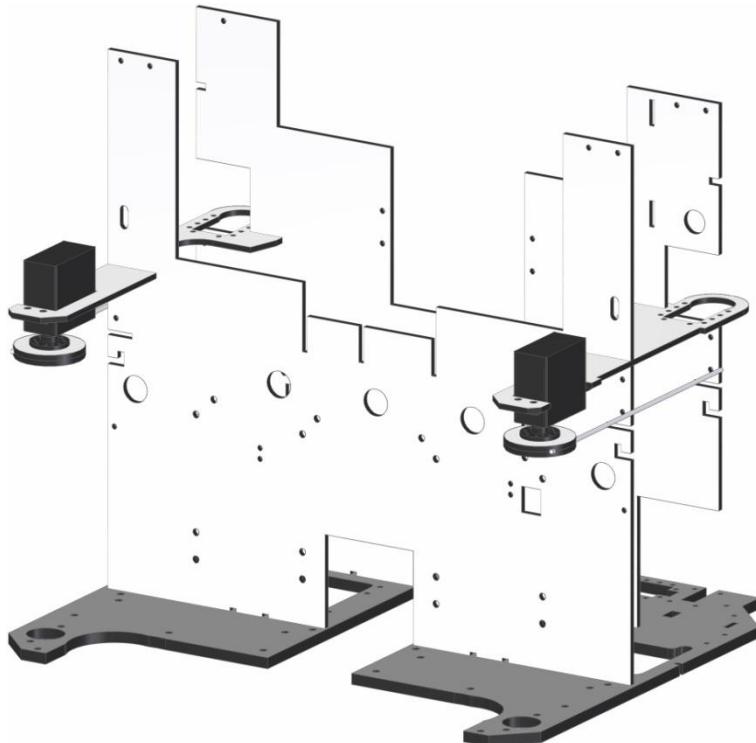


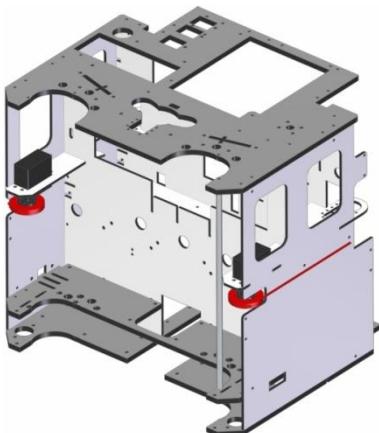
Figura 6-1 *Sistema cierre de claquetas y estructura portante de sustentación*

Continuando la estructuración de los capítulos por sistemas mecánicos de manipulación por elementos de juego, se ha creado este pequeño capítulo para la descripción de un sistema que no ha supuesto ninguna dificultad ni de diseño ni constructiva.

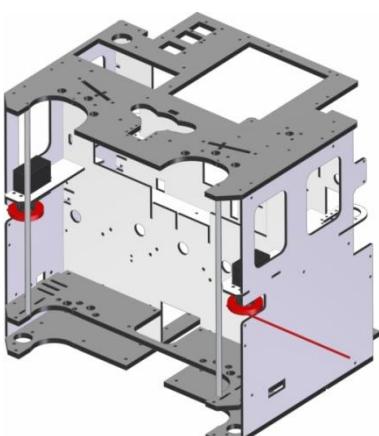
La principal característica definida en las especificaciones ha sido crear un sistema simétrico que permitirá una fácil programación de movimientos, dependiendo del campo por el que se jugara. Este aspecto a tomado una especial relevancia debido a que el sistema diseñado aunque permite por si mismo el cierre de la claqueta, se va a combinar con un movimiento de giro del robot para optimizar el tiempo de ejecución. El cierre se realizará durante un giro para el cambio de trayectoria.

En la Figura 6-1 se muestra el sistema diseñado junto con la estructura portante que le da soporte. Basicamente consiste en dos servos Futaba 3003S con un plato unido a sus propios platos, que mediante una ranura permite la sujeción de unas varillas de fibra de vidrio de dos milímetros de diámetro. Se ha utilizado una varilla de vidrio en lugar de una estructura de más rigidez para evitar una posible rotura del sistema en el caso de pegar contra otro robot durante estas maniobras. La varilla aunque muy flexible permite el cierre de las claquetas y por otro lado es capaz de doblarse evitando su rotura o la propia del servo

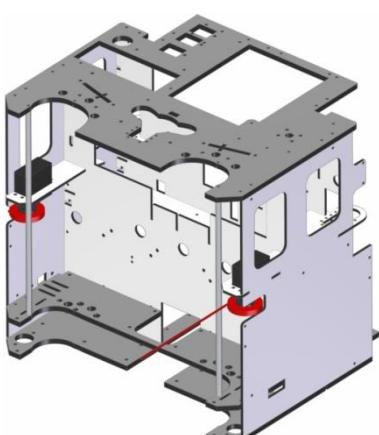
A continuación se muestra la secuencia de movimientos del brazo de cierre de la claqueta entre sus posiciones de giro límites.



Brazo de claqueta derecho replegado.



Brazo de claqueta en posición de 90°



Brazo de claqueta en posición de apertura máxima (180°)

Figura 6-2 Brazo cierre claqueta plegado

Figura 6-3 Brazo cierre claqueta giro a 90°

Figura 6-4 Brazo cierre claqueta a 180°. Angulo abertura máxima

# 7. Análisis de diseño global

---

En este capítulo se recogen algunas de las recopilaciones de datos requeridas por las especificaciones de diseño.

## 7.1. Estimación de masa y posición de centro de masas

---

En este apartado se ha recopilado la información utilizada para la estimación de la masa final del diseño y de la posición del centro de masas a través del software de diseño 3D utilizado. En la Tabla 7-1 se recopilan todos los elementos incluidos en el diseño creado. Cada elemento tiene asignado un material de construcción, con una densidad definida. Existen componentes especiales compuestos, como por ejemplo los servos AX12, para estos se ha definido una masa de usuario, asignándoles el peso indicado por el fabricante.

PIEZA	MATERIAL	DENSIDAD m <sup>3</sup>	MASA kg	VOLUMEN mm <sup>3</sup>	SUPERFICIE mm <sup>2</sup>
Placa Potencia.Par	Compacto alta presión	1300	0,063	48733	30229
Placa Principal.Par	Compacto alta presión	1300	0,047	36450	38406
Torretas.Par	Acero	7833	0,001	177	239
Wt11.1.Par	PVC	1385	0,002	1289	1268
Reductora Grande Gp22A-2St.Par	Acero	7833	0,332	42357	9924
Acetato Base.Par	ABS Plástico, alto impacto	1024	0,005	5130	20731
Acetato Derecha.Par	ABS Plástico, alto impacto	1024	0,005	5056	20478
Bisagra Exterior Balancín Encoder.Par	Compacto alta presión	1300	0,031	23633	17584
Bisagra Interior Balancín Encoder.Par	Compacto alta presión	1300	0,080	61658	24132
Brazo Exterior Balancín Encoder.Par	Compacto alta presión	1300	0,017	13079	10176
Brazo Interior Balancín Encoder.Par	Compacto alta presión	1300	0,014	10902	8840
Conjunto Brazo Encoder Izquierdo.Asm			0,191	83047	
Conjunto Brazo Encoder Izquierdo_Mir.Asm			0,191	83047	
Eje Brazos Encoder.Par	Aluminio, 5050	2684	0,013	4662	1841
Rodamiento Balancin.Par	ABS Plástico, alto impacto	1024	0,000	445	495
Separador Brazos Encoder.Par	Aluminio, 5050	2684	0,005	1767	1090
Separadores Bisagra Balancín Encoder.Par	ABS Plástico, alto impacto	1024	0,002	2111	1539
1.1.4.Par	Acero	7833	0,006	749	955
1.6.Par	Acero	7833	0,009	1113	1351
Baliza V2.Asm			0,649	308949	
Base.1.Par	Compacto alta presión	1300	0,044	33879	13040
Base.Par	Compacto alta presión	1300	0,048	36682	15034
Bumper.Par	PVC	1385	0,002	1146	924
Eedefau0420332316840.Par	Acero	7833	0,017	2180	1866

## Análisis de diseño global

Iso 10511 Nut M6 Acero Prevailing Torque Type Hexagon Thin.3.Par	Acero	7833	0,003	379	416
Iso 4762 Screw M2X3 Acero Hexagon Socket Head Cap.Par	Acero	7833	0,000	29	68
Iso 7089 Washer 6X12 Acero Grade A Plain Normal Series.Par	Acero	7833	0,001	124	243
L0-6882Z.Par	Acero inoxidable	7750	0,002	263	584
Motor Maxon U06079 Con Reductora.Par	Acero	7833	0,193	24692	6823
Part1.1.Par	Hierro	6920	0,038	5443	2931
Part1.6.Par	Compacto alta presión	1300	0,040	30717	15528
Part2.Par	Compacto alta presión	1300	0,010	8061	6207
Part3.9.Par	Compacto alta presión	1300	0,089	68825	24095
Part4.Par	Compacto alta presión	1300	0,054	41219	7541
Part5.Par	Compacto alta presión	1300	0,000	146	297
Part6.Par	Compacto alta presión	1300	0,007	5404	7220
Part7.Par	Compacto alta presión	1300	0,003	2591	3676
Part8.Par	Compacto alta presión	1300	0,003	2303	3242
Part9.Par	Compacto alta presión	1300	0,002	1270	1711
Pletina Bumper Con Rueda.Par	Acero	7833	0,000	33	240
Product24.1.Asm			0,002	1245	
Pz2-41.Par	PVC	1385	0,009	6346	2519
Rueda Bumper.Par	PVC	1385	0,000	66	98
Wt11.Par	PVC	1385	0,002	1289	1268
Base 1.Par	Compacto alta presión	1300	0,403	309718	127620
Bisagra Hembra.Par	Acero inoxidable	7750	0,001	190	689
Bisagra Macho.Par	Acero inoxidable	7750	0,001	183	653
1-St_8X25.Par	Acero	7833	0,509	64964	11109
1.1.5.Par	Acero	7833	0,046	5843	3393
1.Par	Aluminio, 1060	2712	0,068	25115	7782
Bg40X50_Oel_Ol.Par	Acero	7833	0,989	126273	16623
Bloque Motor.Asm			4,279	986198	
Bloque Ruedas.Asm			0,273	152300	
Controladora Bge40.Par	PVC	1385	0,059	42349	11271
Correa Traccion.Par	Silicona	1110	0,006	5224	5587
Encoder Hengstler Ri41.Par	Aluminio, 1060	2712	0,061	36184	6994
Encoder Hengstler Ri41_Mir.Par	ABS Plástico, alto impacto	1024	0,061	36184	6994
Iso 7047 Screw M3X8-H Acero Grade A Countersunk Raised Head.Par	Acero	7833	0,001	70	131
Llanta.Par	PVC	1385	0,097	70114	10323
Motor Dunkermotoren.Asm			1,558	233727	
Placa Encoder Izquierdo.Par	Aluminio, 5050	2684	0,135	50360	24672
Placa Motor Izquierda.Par	Aluminio, 5050	2684	0,123	45646	21896
Rodamiento.Par	Acero inoxidable	7750	0,026	3324	1826
Rueda Encoder.Par	Aluminio, 5050	2684	0,030	11151	5607
Rueda Foam Ancho 40.Par	Silicona	1110	0,056	50423	15967
Disco Brazo Claveta.Par	Compacto alta presión	1300	0,004	3114	2697
Varilla Brazo Claveta.Par	Compacto alta presión	1300	0,001	543	1093
Base Brazos Palomitas 2.Par	Compacto alta presión	1300	0,012	8967	7801
Conjunto Brazo Palomitas Derecho.Par	Compacto alta presión	1300	0,036	27477	20853
Conjunto Brazos			0,036	27477	

Palomitas.Asm					
Estructura Soporte Servo	Compacto alta presión	1300	0,020	15017	12837
Dynamixel Brazos					
Palomitas Derecho.Par					
Estructura Soporte Servo	Compacto alta presión	1300	0,024	18609	15082
Dynamixel Brazos					
Palomitas.Par					
Pendiente Brazos Palomitas	Compacto alta presión	1300	0,003	2169	2229
2.Par					
Suplemento Servo Brazos	Compacto alta presión	1300	0,004	2794	1998
Palomitas.Par					
Tapa Brazos Palomitas 2.Par	Compacto alta presión	1300	0,018	14173	10586
Brazo Carro Horizontal.Par	Compacto alta presión	1300	0,004	3042	2613
Correa Carro	Silicona	1110	0,004	3849	8980
Horizontal.Par					
Guía Carro Horizontal.Par	Aluminio, 1060	2712	0,038	13969	13035
Patín Carro Horizontal.Par	Aluminio, 1060	2712	0,008	3079	2674
Soporte Motor Carro	Compacto alta presión	1300	0,007	5603	3644
Horizontal.Par					
Tensor Inferior Correa.Par	Compacto alta presión	1300	0,006	4672	2831
Cinta Raso Cerrada.Par	ABS Plástico, alto impacto	1024	0,001	549	1995
Alojamiento Eje Servo.Par	Bronce, 90%	8800	0,015	1689	1208
Base 1 Carro Pinza.Par	Compacto alta presión	1300	0,015	11176	8896
Base 2 Carro Pinza.Par	Compacto alta presión	1300	0,007	5401	4843
Base 3 Carro Pinza.Par	Compacto alta presión	1300	0,010	7323	6201
Base 4 Carro Pinza.Par	Compacto alta presión	1300	0,037	28394	10862
Base 4 Carro	Compacto alta presión	1300	0,037	28394	10862
Pinza_Mir1.Par					
Base Carro Servo	Compacto alta presión	1300	0,039	30372	11443
Derecha.Par					
Base Carro Servo	Compacto alta presión	1300	0,039	30372	11443
Derecha_Mir1.Par					
Base Carros Izquierda.Par	Compacto alta presión	1300	0,144	110584	48329
Cabeza Servo Carro	Compacto alta presión	1300	0,002	1809	1724
Pinza.Par					
Carro Pinza 2 Derecha.Asm			0,298	142428	
Carro Pinza 2			0,298	142428	
Izquierda.Asm					
Carro Servo 2 Derecha.Asm			0,322	169846	
Carro Servo 2			0,322	169846	
Izquierda.Asm					
Conjunto Plato Carro			0,023	18053	
Servo.Asm					
Guías 6 Stand.Par	Acero	7833	0,058	7365	4967
Guías 8 Carros.Par	Acero	7833	0,103	13094	6648
Palanca Carro Servo.Par	Compacto alta presión	1300	0,006	4682	3957
Pinza Leva.Par	Compacto alta presión	1300	0,001	734	748
Pinza Palanca.Par	Compacto alta presión	1300	0,004	2780	2523
Pinza Suplemento.Par	Compacto alta presión	1300	0,006	4727	2683
Plato Carro Servo Lado	Compacto alta presión	1300	0,011	8243	6268
Servo.Par					
Plato Carro Servo Lado	Compacto alta presión	1300	0,011	8364	6311
Soporte Eje.Par					
Separador Central Base 1	Aluminio, 5050	2684	0,005	1810	1106
Carro Pinza.Par					
Separador Central Base 2	Aluminio, 5050	2684	0,002	905	653
Carro Pinza.Par					
Separador Central Base 3	Aluminio, 5050	2684	0,010	3563	1839
Carro Pinza.Par					
Separador Central Carro	Aluminio, 5050	2684	0,025	9224	4064
Servo.Par					
Separador Lateral Base 1	Aluminio, 5050	2684	0,002	919	837
Carro Pinza.Par					

## Análisis de diseño global

Separador Lateral Base 2	Aluminio, 5050	2684	0,001	459	470
Carro Pinza.Par					
Separador Lateral Base 3	Aluminio, 5050	2684	0,005	1881	1419
Carro Pinza.Par					
Separador Lateral Carro	Aluminio, 5050	2684	0,018	6528	3567
Servo.Par					
Separador Platos Carro	Compacto alta presión	1300	0,002	1445	1081
Servo.Par					
Soporte Eje Servo Carro	Compacto alta presión	1300	0,020	15202	11255
Servo.Par					
Soporte Servo Carro	Compacto alta presión	1300	0,016	12475	9945
Servo.Par					
Tapa Bisagra Superior.Par	Compacto alta presión	1300	0,010	7502	4534
Tapa Carro Servo	Compacto alta presión	1300	0,039	30372	11443
Derecha.Par					
Tapa Carro Servo	Compacto alta presión	1300	0,039	30372	11443
Derecha_Mir1.Par					
Rodamiento Lineal.Par	Acero inoxidable	7750	0,023	3003	2047
Dynamixel Simplificado.Par	(Asignada masa s/peso)	1146	0,054	47134	12810
Electronica.Asm			0,116	87938	
Sensor Retroreflectivo.Par	PVC	1385	0,009	6374	2495
Separador Soporte	Aluminio, 5050	2684	0,010	3863	2430
Electronica.Par					
Seta De Emergencia.Par	PVC	1385	0,066	47913	12422
Soporte Electronica.Par	Compacto alta presión	1300	0,038	29429	22481
Tapa Electronica.Par	Compacto alta presión	1300	0,183	140558	51439
Tapa Trasera	Aluminio, 1060	2712	0,025	9266	37558
Electronica.Psm					
Tarjeta Interruptores.Par	PVC	1385	0,009	6385	8686
Estructura Chaflán	Compacto alta presión	1300	0,016	12587	9799
Trasero.Par					
Estructura Chaflán	Compacto alta presión	1300	0,016	12587	9799
Trasero_Mir.Par					
Estructura Lateral	Compacto alta presión	1300	0,195	149839	106062
Derecho.Par					
Estructura Lateral	Compacto alta presión	1300	0,196	151057	106541
Izquierdo.Par					
Estructura Pared Vaso.Par	Compacto alta presión	1300	0,018	13665	10004
Estructura Pendiente Piso	Compacto alta presión	1300	0,004	3382	2740
Trasero 2.Par					
Estructura Piso Trasero	Compacto alta presión	1300	0,060	46286	34312
1.Par					
Estructura Piso Trasero	Compacto alta presión	1300	0,067	51532	38486
2.Par					
Estructura Soporte Servo	Compacto alta presión	1300	0,015	11636	9522
Brazos Palomitas					
Derecho.Par					
Estructura Soporte Servo	Compacto alta presión	1300	0,017	13147	10478
Brazos Palomitas.Par					
Estructura Tapa Bloque	Compacto alta presión	1300	0,239	183821	49555
Motor.Par					
Estructura Vertical	Compacto alta presión	1300	0,249	191797	135897
Delantero.Par					
Estructura Vertical	Compacto alta presión	1300	0,242	186179	132077
Trasero.Par					
Futaba 3003.Par	(Asignada masa s/peso)	1120	0,037	33032	7922
Delantero.Par	Compacto alta presión	1300	0,104	79651	15389
Suplemento Trasero.Par	Compacto alta presión	1300	0,021	16208	12656
Trasero Superior.Par	Compacto alta presión	1300	0,168	129253	27164
Trasero.Par	Compacto alta presión	1300	0,200	153565	26814
Palanca Apertura	Compacto alta presión	1300	0,002	1432	1399
Trampillas Inferiores.Par					
Brida Paletas.Par	Compacto alta presión	1300	0,006	4634	2362
Conjunto Paletas.Asm			0,308	220378	

Análisis de diseño global

Correa Paletas Stand.Par	ABS Plástico, alto impacto	1024	0,001	1082	2525
Inferior Estructura Paletas.Par	Compacto alta presión	1300	0,062	47335	18968
Medio Estructura Paletas.Par	Compacto alta presión	1300	0,018	13526	10644
Paleta.Par	Compacto alta presión	1300	0,010	7393	6163
Polea T25 30 Mecanizada.Par	Aluminio, 1060	2712	0,011	4099	2362
Polea T25 30.Par	Aluminio, 1060	2712	0,014	5196	2997
Separador Estructura Di.Par	Aluminio, 5050	2684	0,019	7015	2488
Separador Estructura Ds.Par	Aluminio, 5050	2684	0,010	3629	1521
Separador Estructura Ti.Par	Aluminio, 5050	2684	0,011	4100	1923
Separador Estructura Ts.Par	Aluminio, 5050	2684	0,006	2121	1131
Superior Estructura Paletas.Par	Compacto alta presión	1300	0,020	15035	12685
Apoyo Inferior.Par	Compacto alta presión	1300	0,004	2872	2390
Apoyo Superior.Par	Compacto alta presión	1300	0,011	8846	6696
Brazo Estabilizador Soporte Dynamixel Movil.Par	Compacto alta presión	1300	0,006	4441	3589
Brazo Soporte Dynamixel Movil.Par	Compacto alta presión	1300	0,013	9777	7854
Pinza.Par	Compacto alta presión	1300	0,004	2811	2639
Rampa Base.Par	Compacto alta presión	1300	0,040	30429	23789
Separador Dynamixel Movil.Par	Compacto alta presión	1300	0,008	6132	3902
Soporte Dynamixel Fijo Lateral Derecho.Par	Compacto alta presión	1300	0,041	31612	24975
Soporte Dynamixel Fijo Lateral.Par	Compacto alta presión	1300	0,047	36181	28901
Soporte Dynamixel Fijo.Par	Compacto alta presión	1300	0,026	20308	16520
Pinza Completa.Par	Compacto alta presión	1300	0,007	5765	4962
Pinza Partida Eje.Par	Compacto alta presión	1300	0,005	3592	3129
Pinza Partida Punta.Par	Compacto alta presión	1300	0,003	2417	2193
Polea 25 8Dientes.Par	Aluminio, 1060	2712	0,003	1186	1119
Base Puerta Inferior.Par	Compacto alta presión	1300	0,067	51358	20797
Brazo Puerta Superior.Par	Compacto alta presión	1300	0,007	5446	4593
Conjunto Puerta Derecha.Par	Compacto alta presión	1300	0,037	28504	22218
Conjunto Puerta.Asm			0,037	28504	
Leva Servo Puerta.Par	Compacto alta presión	1300	0,002	1850	1735
Puerta.Par	Compacto alta presión	1300	0,023	17612	13500
Rodamiento Puerta.Par	Acero inoxidable	7750	0,011	1400	975
Soporte Servo Inferior.Par	Compacto alta presión	1300	0,007	5234	4755
Refuerzo Delantero Base.Par	Hierro	6920	0,443	64038	20662
Casquillo Autolubricado 3-6-5.Par	Bronce, 90%	8800	0,001	156	255
Casquillo Autolubricado 4-8-5.Par	Bronce, 90%	8800	0,003	310	397
Roll-On Pequenio.Par	Acero	7833	0,043	5499	1742
Separador Base Y Base Carros Delantero.Par	Aluminio, 5050	2684	0,003	970	878
Separador Lastre Y Base Carros Delantero.Par	Aluminio, 5050	2684	0,003	990	792
Separador Trasero Base Carros Y Base Puerta.Par	Aluminio, 5050	2684	0,005	1715	1275
Soporte Rollon Derecha.Par	Compacto alta presión	1300	0,025	19296	6735
Soporte Rollon.Par	Compacto alta presión	1300	0,025	19296	6735
Soporte Teflón Apoyo Trasero.Par	Compacto alta presión	1300	0,006	4786	2642
Tapa Superior	Compacto alta presión	1300	0,392	301602	124557

Estructura.Par

Taquitos De Aluminio.Par	Aluminio, 1060	2712	0,008	2875	1592
Teflón Apoyo Trasero.Par	ABS Plástico, alto impacto	1024	0,002	1473	1022
Bisagra Trampilla Superior.Par	Compacto alta presión	1300	0,007	5735	2767
Brazo Servo Trampilla Superior 2.Par	Compacto alta presión	1300	0,006	4641	3870
Leva Servo Trampilla Superior 2.Par	Compacto alta presión	1300	0,006	4737	3840
Soporte Servo Trampilla Superior.Par	Compacto alta presión	1300	0,007	5311	4464
Trampilla Trasera Superior.Psm	Aluminio, 1060	2712	0,061	22468	45804
Trampilla Trasera Inferior Derecha.Psm	Aluminio, 1060	2712	0,015	5463	22131
Trampilla Trasera Inferior.Psm	Aluminio, 1060	2712	0,015	5463	22131
Conjunto Robot Centro De Masas.Asm			13,666	6504503	

Tabla 7-1 Cálculo detallado de masa total del diseño

La masa final del diseño estimada es de 13.666 Kg. Este dato ha sido utilizado para comprobar la aceleración máxima del robot y verificar que se cumplía la especificación para esta.

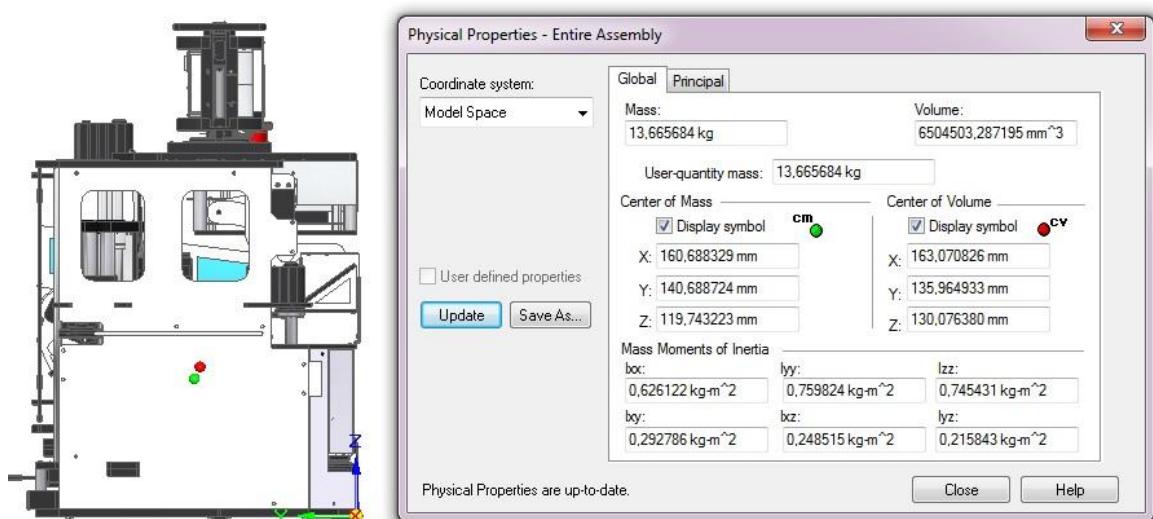


Figura 7-1 Ventana de cálculo de posición de centro de masas de Solid Edge

El software de diseño 3D utilizado también ha permitido obtener la disposición del centro de masas final. Este dato es de crucial importancia en el diseño de este tipo de robots, para calcular la aceleración límite de patinaje de las ruedas o la posibilidad de caballito según la distribución de las ruedas de tracción respecto a los apoyos delanteros o traseros de existir. Esta aceleración límite, menos un porcentaje de error de cálculo, será la empleada como aceleración máxima configurable en las rampas de aceleración y deceleración. En la Figura 7-1 se muestra la ventana de calculo de la posición del centro de masas respecto al sistema de

coordenadas elegido del software de diseño. En este caso los datos obtenidos están referenciados desde un sistema de coordenadas con su origen localizado en el plano de la cara inferior de la base del robot, en el punto inferior derecho de la Figura 7-2. En esta figura se han acotado todos los datos necesarios para los cálculos a realizar. Estos son:

- Distancia del eje motriz al centro de masas -0.01918m
- Distancia del centro de masas al eje de apoyo trasero 0.1397m
- Distancia del centro de masas al eje de apoyo trasero 0.11182m
- Altura del centro de masas 0.1229
- Coeficiente de aderencia 1

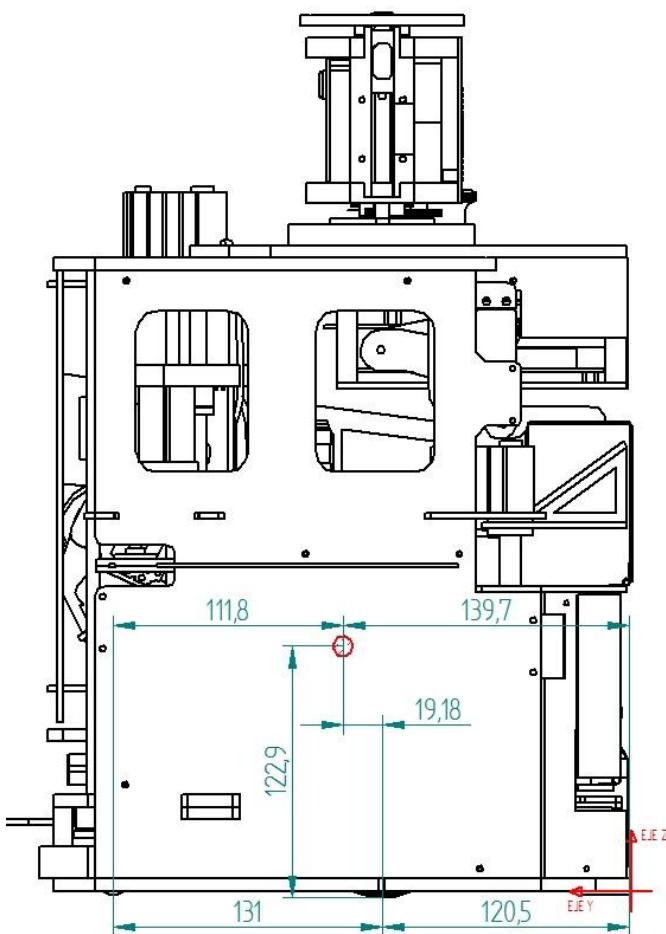


Figura 7-2 Localización de centro de masas y disposición de apoyos delanteros y traseros

Antes de continuar con los cálculos es necesario aclarar algunos conceptos que van a ser utilizados y que toman como base un trabajo publicado por el equipo de competición de Eurobot RCVA y cuyo autor es Jacques Coulon en Otoño de 2007. Todos los cálculos realizados a continuación tienen su fundamento en este documento.

## ESTIMACIÓN DEL COEFICIENTE DE ADERENCIA DE LAS RUEDAS UTILIZADAS

El valor utilizado para los cálculos supone que el valor conseguido en el diseño será mayor a 1, utilizando como base los valores obtenidos en los desarrollos realizados para las anteriores participaciones en esta prueba. El cálculo de este valor se realiza mediante un método que consiste en comprobar la fuerza límite necesaria para desplazar el robot traccionando horizontalmente de él sobre una superficie plana y con el acabado de rugosidad de los campos sobre los que se competirá. Para calcularlo basta con una cuerda atada a la base del robot y una polea para dirigir horizontalmente la fuerza de un cubo cargado con un peso determinado. También es preciso pesar con una báscula el robot en el punto de apoyo de las ruedas de tracción. Obtenidos estos dos datos el cálculo es el siguiente:

$$k_{aderencia} = \frac{\text{Peso límite antes de iniciarse el deslizamiento}}{\text{Peso del robot sobre las ruedas de tracción}}$$

Con unas ruedas espumosas forradas de una delgada capa siliconada en anteriores diseños los valores se acercaban a 1,07. El peso necesario para desplazar el robot era mayor que el transmitido por las ruedas al piso.

## CALCULO DE ACELERACIÓN LÍMITE PARA EL PATINAJE DE LAS RUEDAS

La aceleración límite como se demuestra en el documento de referencia citado no depende del peso del robot, solo de la disposición del centro de masas en la estructura y el coeficiente de adherencia. El cálculo realizado es el siguiente.

$$a_{MAX}(g) = \frac{k_{aderencia} \times d_{CM \rightarrow ApoyoTrasero}}{(d_{CM \rightarrow ApoyoTrasero} + d_{Rueda \rightarrow CM}) + (Z_{CM} \times d_{CM \rightarrow ApoyoTrasero})}$$

Para la dirección de avance seria:

$$a_{MAX}(g) = \frac{1 \times 0.1397}{0.1397 + (-0.01918) + (0.1229 \times 0.1397)} = \frac{0.1397}{0.13768913} = 1.014(g)$$

$$a_{MAX}(m/s^2) = 1.014(g) \times 9.8 \left( \frac{m/s^2}{g} \right) = 9.94(m/s^2)$$

Para la dirección de retroceso seria:

$$a_{MAX}(g) = \frac{1 \times 0.11182}{0.11182 + 0.01918 + (0.1229 \times 0.11182)} = \frac{0.11182}{0.144742678} = 0.772(g)$$

$$a_{MAX}(m/s^2) = 0.772(g) \times 9.8 \left( \frac{m/s^2}{g} \right) = 7.57(m/s^2)$$

En el caso del retroceso la aceleración límite es menor por la menor distancia del centro de masas al punto de apoyo delantero y por encontrarse la rueda por delante del centro de masas en la dirección del avance.

Aunque a los valores calculados se les aplica una reducción de un 20%, se puede concluir que para ambos casos la aceleración límite se encuentra muy por encima del valor que se espera conseguir según las especificaciones de diseño y que se estableció en  $3.6 \text{ m/s}^2$ .

## 7.2. Listado piezas mecanizables

En este apartado se incluyen los listados de las diferentes piezas del diseño agrupados en los siguientes conjuntos:

- Piezas mecanizadas de materiales fenólicos. Tabla 7-2
- Piezas mecanizadas de metal. Tabla 7-2
- Piezas de torno. Tabla 7-4
- Componentes mecánica. Tabla 7-5
- Componentes electrónica. Tabla 7-6

Todas las listas creadas están ordenadas por conjuntos y subconjuntos del diseño y por número de plano. Se indica nombre, material, cantidad y proveedor.

### *Listado piezas mecanizadas de materiales fenólicos*

Conjunto	SubConjunto	Nº Plano	Descripción	Material	Cantidad	Proveedor
ESTRUCTURA ROBOT						
Robot	Estructura	00-01-01-001	Base 1	Trespa 6	1	Carpintería Ingles
Robot	Estructura	00-01-02-014	Tapa Superior Estructura	Trespa 6	1	Carpintería Ingles
Robot	Estructura	00-01-03-007	Estructura Vertical Delantero	Trespa 3	1	Carpintería Ingles
Robot	Estructura	00-01-04-008	Estructura Vertical Trasero	Trespa 3	1	Carpintería Ingles
Robot	Estructura	00-01-05-010	Estructura Lateral Izquierdo	Trespa 3	1	Carpintería Ingles
Robot	Estructura	00-01-06-011	Estructura Lateral Derecho	Trespa 3	1	Carpintería Ingles
Robot	Estructura	00-01-07-025	Estructura Tapa Bloque Motor	Trespa 10	1	Carpintería Ingles
Robot	Estructura	00-01-08-036	Estructura Soporte Brazos Palomitas	Trespa 3	1	Carpintería Ingles
Robot	Estructura	00-01-09-038	Estructura Soporte Brazos Palomitas Derecho	Trespa 3	1	Carpintería Ingles
Robot	Estructura	00-01-12-012	Estructura Chaflán Trasero	Trespa 3	$1 \text{ S/D} + 1 \text{ M/C}$	Carpintería Ingles
Robot	Estructura	00-01-13-027	Estructura Piso Trasero	Trespa 3	1	Carpintería Ingles
Robot	Estructura	00-01-14-026	Estructura Piso Trasero 1	Trespa 3	1	Carpintería Ingles
Robot	Estructura	00-01-15-028	Estructura Pared Vaso	Trespa 3	2	Carpintería Ingles
Robot	Estructura	00-01-16-029	Estructura Pendiente Piso Trasero 1	Trespa 3	2	Carpintería Ingles
Robot	Estructura	00-01-17-089	Palanca Apertura Trampillas Inferiores	Trespa 3	2	Carpintería Ingles
Robot	Estructura	00-01-18-203	Soporte Teflón Apoyo Trasero	Trespa 6	2	Carpintería Ingles

## Análisis de diseño global

Robot	Estructura	00-01-22-050	Soporte Rollon	Trespa 16	1 S/D + 1 M/C	Carpintería Ingles
LASTRE						
Lastre	Estructura	02-00-04-204	Suplemento Trasero	Trespa 3	1	Carpintería Ingles
BALANCINENCODER						
Balancín Encoder	Estructura	03-00-01-030	Bisagra Exterior Balancín Encoder	Trespa 3	2	Carpintería Ingles
Balancín Encoder	Estructura	03-00-02-031	Bisagra Interior Balancín Encoder	Trespa 3	2	Carpintería Ingles
Balancín Encoder	Estructura	03-00-03-032	Brazo Exterior Balancín Encoder	Trespa 3	2	Carpintería Ingles
Balancín Encoder	Estructura	03-00-04-033	Brazo Interior Balancín Encoder	Trespa 3	2	Carpintería Ingles
CONJUNTO CARRO 2						
Conjunto Carro 2	Estructura	04-00-01-009	Base Carros Izquierda	Trespa 6	1	Carpintería Ingles
Conjunto Carro 2	Estructura	04-00-02-093	Tapa Bisagra Superior	Trespa 6	2	Carpintería Ingles
Conjunto Carro 2	Carro Pinza	04-01-01-106	Base 1 Carro Pinza	Trespa 3	2	Carpintería Ingles
Conjunto Carro 2	Carro Pinza	04-01-02-107	Base 3 Carro Pinza	Trespa 3	2	Carpintería Ingles
Conjunto Carro 2	Carro Pinza	04-01-03-108	Base 2 Carro Pinza	Trespa 3	2	Carpintería Ingles
Conjunto Carro 2	Carro Pinza	04-01-04-109	Base 4 Carro Pinza	Trespa 10	1 S/D + 1 M/C	Carpintería Ingles
Conjunto Carro 2	Carro Pinza	04-01-05-110	Pinza Suplemento	Trespa 6	8	Carpintería Ingles
Conjunto Carro 2	Carro Pinza	04-01-06-111	Pinza Palanca	Trespa 3	4	Carpintería Ingles
Conjunto Carro 2	Carro Pinza	04-01-07-112	Pinza Leva	Trespa 3	8	Carpintería Ingles
Conjunto Carro 2	Carro Pinza	04-01-08-113	Cabeza Servo Carro Pinza	Trespa 3	2	Carpintería Ingles
Conjunto Carro 2	Carro Pinza	04-01-09-114	Alojamiento Eje Servo	Trespa 6	2	Carpintería Ingles
Conjunto Carro 2	Carro Servo	04-02-01-096	Palanca Carro Servo	Trespa 3	4	Carpintería Ingles
Conjunto Carro 2	Carro Servo	04-02-02-097	Plato Carro Servo Lado Soporte Eje	Trespa 3	2	Carpintería Ingles
Conjunto Carro 2	Carro Servo	04-02-03-098	Plato Carro Servo Lado Servo	Trespa 3	2	Carpintería Ingles
Conjunto Carro 2	Carro Servo	04-02-04-099	Separador Platos Carro Servo	Trespa 5	2	Carpintería Ingles
Conjunto Carro 2	Carro Servo	04-02-05-100	Tapa Carro Servo Derecha_Mir1	Trespa 10	1 S/D + 1 M/C	Carpintería Ingles
Conjunto Carro 2	Carro Servo	04-02-06-101	Base Carro Servo Derecha_Mir1	Trespa 10	1 S/D + 1 M/C	Carpintería Ingles
Conjunto Carro 2	Carro Servo	04-02-07-102	Soporte Servo Carro Servo	Trespa 3	2	Carpintería Ingles
Conjunto Carro 2	Carro Servo	04-02-08-103	Soporte Eje Servo Carro Servo	Trespa 3	2	Carpintería Ingles
PALETAS STAND						
Paletas Stand	Estructura	05-00-01-002	Inferior Estructura Paletas	Trespa 6	1	Carpintería Ingles
Paletas Stand	Estructura	05-00-02-034	Medio Estructura Paletas	Trespa 3	1	Carpintería Ingles
Paletas Stand	Estructura	05-00-03-035	Superior Estructura Paletas	Trespa 3	1	Carpintería Ingles
Paletas Stand	Estructura	05-00-09-063	Paleta	Trespa 3	2	Carpintería Ingles
CARRO HORIZONTAL						
Carro Horizontal	Estructura	06-00-02-051	Brazo Carro Horizontal	Trespa 3	1	Carpintería Ingles
Carro Horizontal	Estructura	06-00-03-067	Soporte Motor Carro Horizontal	Trespa 6	1	Carpintería Ingles
Carro Horizontal	Estructura	06-00-04-070	Tensor Inferior Correa	Trespa 10	2	Carpintería Ingles
PUERTA STAND						
Puerta Stand	Estructura	07-00-01-020	Base Puerta Inferior	Trespa 6	1	Carpintería Ingles

Puerta Stand	Estructura	07-00-02-047	Brazo Puerta Superior	Trespa 3	4	Carpintería Ingles
Puerta Stand	Estructura	07-00-03-048	Puerta	Trespa 3	2	Carpintería Ingles
Puerta Stand	Estructura	07-00-04-079	Leva Servo Puerta	Trespa 3	2	Carpintería Ingles
Puerta Stand	Estructura	07-00-05-200	Soporte Servo Inferior	Trespa 3	1	Carpintería Ingles
PINZA VASO DELANTERO						
Pinza Vaso Delantero	Estructura	08-00-01-021	Rampa Base	Trespa 3	1	Carpintería Ingles
Pinza Vaso Delantero	Estructura	08-00-02-022	Soporte Dynamixel Fijo Lateral	Trespa 3	1	Carpintería Ingles
Pinza Vaso Delantero	Estructura	08-00-03-023	Soporte Dynamixel Fijo Lateral Derecho	Trespa 3	1	Carpintería Ingles
Pinza Vaso Delantero	Estructura	08-00-04-024	Soporte Dynamixel Fijo	Trespa 3	1	Carpintería Ingles
Pinza Vaso Delantero	Brazo	08-01-01-041	Apoyo Inferior	Trespa 3	1	Carpintería Ingles
Pinza Vaso Delantero	Brazo	08-01-02-042	Apoyo Superior	Trespa 3	1	Carpintería Ingles
Pinza Vaso Delantero	Brazo	08-01-03-043	Brazo Soporte Dynamixel Móvil	Trespa 3	1	Carpintería Ingles
Pinza Vaso Delantero	Brazo	08-01-04-044	Brazo Estabilizador Soporte Dynamixel Móvil	Trespa 3	1	Carpintería Ingles
Pinza Vaso Delantero	Brazo	08-01-05-045	Separador Dynamixel Móvil	Trespa 8	1	Carpintería Ingles
Pinza Vaso Delantero	Brazo	08-01-06-046	Pinza	Trespa 3	1	Carpintería Ingles
Brazo Claqueta	Estructura	09-00-01-071	Disco Brazo Claqueta	Trespa 3	2 S/D + 2 M/C	Carpintería Ingles
BRAZOS PALOMITAS						
Brazos Palomitas	Estructura	10-01-01-037	Estructura Soporte Servo Dynamixel Brazos Palomitas	Trespa 3	1	Carpintería Ingles
Brazos Palomitas	Estructura	10-01-02-039	Estructura Soporte Servo Dynamixel Brazos Palomitas Derecho	Trespa 3	1	Carpintería Ingles
Brazos Palomitas	Estructura	10-01-03-083	Base Brazos Palomitas 2	Trespa 3	2	Carpintería Ingles
Brazos Palomitas	Estructura	10-01-04-082	Pendiente Brazos Palomitas 2	Trespa 3	4	Carpintería Ingles
Brazos Palomitas	Estructura	10-01-05-081	Tapa Brazos Palomitas 2	Trespa 3	2	Carpintería Ingles
PINZA VASO						
Pinza Vaso	Estructura	11-00-01-086	Pinza Completa	Trespa 3	1	Carpintería Ingles
Pinza Vaso	Estructura	11-00-02-087	Pinza Partida Eje	Trespa 3	1	Carpintería Ingles
Pinza Vaso	Estructura	11-00-03-088	Pinza Partida Punta	Trespa 3	1	Carpintería Ingles
TRAMPILLA SUPERIOR						
Trampilla Superior	Estructura	12-00-01-075	Leva Servo Trampilla Superior 2	Trespa 3	1	Carpintería Ingles
Trampilla Superior	Estructura	12-00-02-076	Brazo Servo Trampilla Superior 2	Trespa 3	1	Carpintería Ingles
Trampilla Superior	Estructura	12-00-03-077	Bisagra Trampilla Superior	Trespa 10	1	Carpintería Ingles
Trampilla Superior	Estructura	12-00-04-201	Soporte Servo Trampilla Superior	Trespa 3	1	Carpintería Ingles
SOPORTE ELECTRÓNICA						
Electrónica	Estructura	14-00-01-015	Tapa Electrónica	Trespa 6	1	Carpintería Ingles
Electrónica	Estructura	14-00-03-040	Soporte Electrónica	Trespa 3	1	Carpintería Ingles

Tabla 7-2 Listado piezas mecanizadas de materiales fenólicos

*Listado piezas mecanizadas en metal*

Conjunto	SubConjunto	Nº Plano	Descripción	Material	Cantidad	Proveedor
----------	-------------	----------	-------------	----------	----------	-----------

ESTRUCTURA ROBOT						
Robot	Estructura	00-01-10-084	Tacos Aluminio	Macizo Aluminio 10x10	23	Ayudas Hidráulicas S.L.
Robot	Estructura	00-01-11-003	Refuerzo Delantero Base	Acero F-100	1	Ayudas Hidráulicas S.L.
Robot	Cubierta	00-02-01-013	Trampilla Trasera Inferior	Chapa Aluminio 0,5mm	1 S/D + 1 M/C	Ayudas Hidráulicas S.L.
Robot	Cubierta	00-02-02-017	Trampilla Trasera Superior	Chapa Aluminio 0,5mm	1	Ayudas Hidráulicas S.L.
BLOQUE MOTOR						
Bloque Motor	Estructura	01-00-01-000	Placa Encoder Izquierdo	Aluminio 5083 8mm	2	Ayudas Hidráulicas S.L.
Bloque Motor	Estructura	01-00-02-000	Placa Motor Izquierda	Aluminio 5083 8mm	2	Ayudas Hidráulicas S.L.
LASTRE						
Lastre	Estructura	02-00-01-006	Trasero	Acero	1	Ayudas Hidráulicas S.L.
Lastre	Estructura	02-00-02-006	Trasero Superior	Acero	1	Ayudas Hidráulicas S.L.
Lastre	Estructura	02-00-03-005	Delantero	Acero F-100	2	Ayudas Hidráulicas S.L.
CARRO HORIZONTAL						
Carro Horizontal	Estructura	06-00-01-004	Guía Carro Horizontal (mecanizado guía Igus)	Guía Igus Ts-04-09	1	Ayudas Hidráulicas S.L.
SOPORTE ELECTRÓNICA						
Electrónica	Estructura	14-00-02-018	Tapa Trasera Electrónica	Chapa Aluminio 0,5mm	1	Ayudas Hidráulicas S.L.

Tabla 7-3 Listado piezas mecanizadas de metal

*Listado piezas torno*

Conjunto	SubConjunto	Nº Plano	Descripción	Material	Cantidad	Proveedor
ESTRUCTURA ROBOT						
Robot	Estructura	00-01-19-054	Soporte Trasero Base Carros	Aluminio DuralØ10	4	Ayudas Hidráulicas S.L.
Robot	Estructura	00-01-20-055	Soporte Lastre Y Base Carros Delantero	Aluminio DuralØ10	4	Ayudas Hidráulicas S.L.
Robot	Estructura	00-01-21-053	Soporte Base Y Base Carros Delantero	Aluminio DuralØ9	2	Ayudas Hidráulicas S.L.
BLOQUE MOTOR						
Bloque Motor	Estructura	01-00-03-000	Llanta	Aluminio Dural	2	Ayudas Hidráulicas S.L.
BALANCINENCODER						
Balancín Encoder	Estructura	03-00-05-000	Rueda Balancín Encoder	Aluminio DuralØ53	2	Ayudas Hidráulicas S.L.
Balancín Encoder	Estructura	03-00-06-065	Separador Brazos Encoder	Aluminio DuralØ10	8	Ayudas Hidráulicas S.L.
Balancín Encoder	Estructura	03-00-07-066	Eje Brazos Encoder	Aluminio DuralØ14	8	Ayudas Hidráulicas S.L.
CONJUNTO CARRO 2						
Conjunto Carro 2	Estructura	04-00-03-094	Guías 6 Stand	Varilla Acero Plata Ø6	4	Ayudas Hidráulicas S.L.
Conjunto Carro 2	Estructura	04-00-04-095	Guías 8 Carros	Varilla Acero Plata Ø8	4	Ayudas Hidráulicas S.L.
Conjunto Carro 2	Carro Pinza	04-01-10-115	Separador Lateral Base 1 Carro Pinza	Aluminio DuralØ9	4	Ayudas Hidráulicas S.L.
Conjunto Carro 2	Carro Pinza	04-01-11-116	Separador Central Base 1 Carro Pinza	Aluminio DuralØ12	4	Ayudas Hidráulicas S.L.
Conjunto Carro 2	Carro Pinza	04-01-12-117	Separador Lateral Base 2 Carro Pinza	Aluminio DuralØ9	4	Ayudas Hidráulicas S.L.
Conjunto Carro 2	Carro Pinza	04-01-13-118	Separador Central Base 2 Carro Pinza	Aluminio DuralØ12	2	Ayudas Hidráulicas S.L.
Conjunto Carro 2	Carro Pinza	04-01-14-119	Separador Lateral Base 3 Carro Pinza	Aluminio DuralØ9	4	Ayudas Hidráulicas S.L.
Conjunto Carro 2	Carro Pinza	04-01-15-120	Separador Central Base 3 Carro Pinza	Aluminio DuralØ12	2	Ayudas Hidráulicas S.L.
Conjunto Carro 2	Carro Servo	04-02-09-104	Separador Lateral Carro Servo	Aluminio DuralØ12	4	Ayudas Hidráulicas S.L.
Conjunto Carro 2	Carro Servo	04-02-10-105	Separador Central Carro Servo	Aluminio DuralØ14	2	Ayudas Hidráulicas S.L.
PALETAS STAND						
Paletas Stand	Estructura	05-00-04-056	Separador Estructura Ti	Aluminio DuralØ14	1	Ayudas Hidráulicas S.L.
Paletas Stand	Estructura	05-00-05-057	Separador Estructura Ts	Aluminio DuralØ14	1	Ayudas Hidráulicas S.L.
Paletas Stand	Estructura	05-00-06-058	Separador Estructura Di	Aluminio DuralØ18	1	Ayudas Hidráulicas S.L.
Paletas Stand	Estructura	05-00-07-059	Separador Estructura Ds	Aluminio DuralØ18	1	Ayudas Hidráulicas S.L.
Paletas Stand	Estructura	05-00-08-062	Brida Paletas	Aluminio DuralØ26	1	Ayudas Hidráulicas S.L.
BRAZO CLAQUETA						
Brazo Claqueta	Estructura	09-00-02-000	Varilla Brazo Claqueta	Fibra De Vidrio Ø2	2	Ayudas Hidráulicas S.L.
BRAZOS PALOMITAS						
Brazos Palomitas	Estructura	10-01-06-085	Suplemento Servo Brazos Palomitas	Aluminio DuralØ22	2	Ayudas Hidráulicas S.L.
SOPORTE ELECTRÓNICA						
Electrónica	Estructura	14-00-04-078	Separador Soporte Electrónica	Aluminio DuralØ10	4	Ayudas Hidráulicas S.L.

Tabla 7-4 *Listado piezas torno*

*Listado piezas mecánica*

Conjunto	SubConjunto	Nº Plano	Descripción	Material	Cantidad	Proveedor
ESTRUCTURA ROBOT						
Robot	Estructura		Unidad De Transferencia De Bola (Rollon)Altura:19.6mm, Diámetro :20.6mm Diámetro de Bola:12.7mm, Peso de Carga Máx.:35kg (Ref. Farnell 1779545)		2	FarnellComponentsSL.
Robot	Cubierta		Bisagra Recta Hierro Zincado 25x18 mm		6	LeroyMerlin
Robot	Cubierta		Cinta de Raso ancho 10 mm Negra		0,5	Estilo pontejos
BLOQUE MOTOR						
Bloque Motor		01-00-04-000	Rueda Foam Ancho 40	Tubo Foam 7,5mm	2	LeroyMerlin
Bloque Motor	Estructura		Rodamiento GPZExt Ø26 , Int Ø10, Esp 8mm		4	WorldBearing S.L.
Bloque Motor	Transmisión		POLEA DENTADA PASO 5MM30 DIENTES	Aluminio	2	WorldBearing S.L.
Bloque Motor	Transmisión		POLEA DENTADA PASO 5MM15 DIENTES	Aluminio	2	WorldBearing S.L.
Bloque Motor	Transmisión		CORREA DENTADA SYNCHROBELT PASO 5MM 225 MM	Poliuretano armada con Acero	2	WorldBearing S.L.
BALANCINENCODER						
Balancín Encoder	Estructura		Rodamiento miniatura Ext Ø15 , Int Ø6, Esp 5mm (12696-ZZ)		4	WorldBearing S.L.
CONJUNTO CARRO 2						
Conjunto Carro 2	Carro Pinza		Rodamiento lineal Ext Ø15 , Int Ø8, Longitud 24mmLM8UU 8mm Linear Bearing	Acero al carbono	8	(AliExpress) Tigoo Technology Co., Ltd.  China (Mainland)
Conjunto Carro 2	Carro Pinza		Casquillos sinterizados en bronce autolubricado TIPO B Ext Ø8 , Int Ø4, Valona Ø12, Longitud 4 mm, Espesor Valona 2 mm	Bronce sinterizado	8	Maymol S.A.
Conjunto Carro 2	Carro Pinza		Casquillos sinterizados en bronce autolubricado TIPO B Ext Ø6 , Int Ø3, Valona Ø9, Longitud 4 mm, Espesor Valona 1,5 mm	Bronce sinterizado	2	Maymol S.A.
Conjunto Carro 2	Carro Servo		Casquillos sinterizados en bronce autolubricado TIPO B Ext Ø6 , Int Ø3, Valona Ø9, Longitud 4 mm, Espesor Valona 1,5 mm	Bronce sinterizado	4	Maymol S.A.
Conjunto Carro 2	Carro Servo		Rodamiento miniatura Ext Ø10 , Int Ø4, Esp 4mm (12MR-104-ZZ)		2	WorldBearing S.L.
Conjunto Carro 2	Carro Servo		Rodamiento lineal Ext Ø15 , Int Ø8, Longitud 24mmLM8UU 8mm Linear Bearing	Acero al carbono	8	(AliExpress)Tigoo Technology Co., Ltd.China (Mainland)
PALETAS STAND						
Paletas Stand	Transmisión		POLEA DENTADA PASO 2,5MM30 DIENTES (1016T2.5-Z-30)	Aluminio	2	WorldBearing S.L.
Paletas Stand	Transmisión		CORREA DENTADA PASO 2,5MM177,5MM (26T-2.5-177.5-6)	Poliuretano armada con Acero	1	WorldBearing S.L.
CARRO HORIZONTAL						

Carro Horizontal	Estructura		Guía Igus "DryLin T" Ts-04-09 Longitud 324 mm	Aluminio Tratado	0,32	Igus Polymer Innovations, S.L. Unipersonal
Carro Horizontal	Estructura		Carro deslizante TW-04-09		1	Igus Polymer Innovations, S.L. Unipersonal
Carro Horizontal	Transmisión		Rodamiento miniatura Ext Ø10, Int Ø4, Esp 4mm (12MR-104-ZZ)		2	WorldBearing S.L.
Carro Horizontal	Transmisión		POLEA DENTADA PASO 2,5MM12 DIENTES (1016T2.5-Z-12)	Aluminio	2	WorldBearing S.L.
Carro Horizontal	Transmisión		CORREA DENTADA PASO 2,5MM650MM (26T-2.5-650-6)	Poliuretano armada con Acero	1	WorldBearing S.L.
<b>PUERTA STAND</b>						
Puerta Stand	Estructura		Rodamiento GPZExt Ø19 , Int Ø8, Esp 6mm (01626-ZZ)		4	WorldBearing S.L.
Puerta Stand	Estructura		Cordon elástico Ø3 Negro		1 m	Estilo poncejos

Tabla 7-5 Listado piezas mecánica

*Listado piezas electrónica*

Conjunto	SubConjunto	Nº Plano	Descripción	Material	Cantidad	Proveedor
<b>ESTRUCTURA ROBOT</b>						
Robot	Electrónica		Sensor Omron E3Z-D61		9	EBAY
<b>BLOQUE MOTOR</b>						
Blóque Motor	Electrónica		Motor brushlessDunkermotor en BG40x50con reductora incorporada PLG42S		2	EBAY
<b>BALANCINENCODER</b>						
Balancín Encoder	Estructura		Encoder RI41 Hengstler 3600 pulsos		2	EBAY
<b>" CONJUNTO CARRO 2</b>						
Conjunto Carro 2	Carro Pinza		Futaba S3003 Servo 180°		2	PC COMPONENTES Y MULTIMEDIA SL
Conjunto Carro 2	Carro Servo		Servo Digital Dynamixel AX12-A		2	RO-BOTICA GLOBAL S.L.
<b>PALETAS STAND</b>						
Paletas Stand	Estructura		Servo Digital Dynamixel AX12-A		2	RO-BOTICA GLOBAL S.L.
<b>CARRO HORIZONTAL</b>						
Carro Horizontal	Transmisión		Motor Maxon 2322.980-52.235-200 24V 6W, reductora 14:1, Encoder 100 pulsos 2 canales cuadratura		1	EBAY
<b>PUERTA STAND</b>						
Puerta Stand	Estructura		Servo Digital Dynamixel AX12-A		2	RO-BOTICA GLOBAL S.L.
<b>PINZA VASO DELANTERO</b>						
Pinza Vaso Delantero	Estructura		Servo Digital Dynamixel AX12-A		1	RO-BOTICA GLOBAL S.L.
Pinza Vaso Delantero	Brazo		Servo Digital Dynamixel AX12-A		1	RO-BOTICA GLOBAL S.L.
<b>BRAZO CLAQUETA</b>						

Brazo Claqueta	Estructura		Futaba S3003 Servo 180°		2	PC COMPONENTES Y MULTIMEDIA SL
BRAZOSPALOMITAS						
Brazos Palomitas	Estructura		Servo Digital Dynamixel AX12-A		2	RO-BOTICA GLOBAL S.L.
PINZA VASO						
Pinza Vaso	Estructura		Servo Digital Dynamixel AX12-A		2	RO-BOTICA GLOBAL S.L.
TRAMPILLA SUPERIOR						
Trampilla Superior	Estructura		Futaba S3003 Servo 180°		1	PC COMPONENTES Y MULTIMEDIA SL
SOPORTE ELECTRONICA						
Electrónica			Bolsa de 10 cables de 4 pins de 20 cm para servos Dynamixel. (Ref. 2495)		1	RO-BOTICA GLOBAL S.L.
Electrónica			Bolsa de 10 cables de 4 pins de 24 cm para servos Dynamixel. (Ref. 2490)		1	RO-BOTICA GLOBAL S.L.
Electrónica			Bobina estanó 0.5 mm 250grs sin plomo		1	FarnellComponentsSL.

Tabla 7-6 Listado piezas mecánica

### 7.3. Listado tornillería

En este apartado se detalla la tornillería utilizada entre las uniones de cada una de las piezas diseñadas (Ver Tabla 7-7). En las columnas "Pieza 1" y "Pieza 2" se indican las referencias de las dos piezas a unir, este código aparece en los planos en el último campo y es un número único asignado a cada una de las piezas para las operaciones de montaje.

En la columna "Conjuntos" se indican el número de conjuntos de piezas iguales a unir y en la de "Cantidad/Conjunto" el número de tornillos de ese tipo para un único conjunto. El total de los tornillos de ese tipo se calculará multiplicando ambas columnas. También se indica el DIN la Métrica y la Longitud.

Pieza 1	Pieza 2	Conjuntos	Métrica	Longitud (mm)	Cantidad/Conjunto	DIN
1	50	2	M4	20	2	7991
1	9	4	M4	55	1	7991
50		2	M5	5	1	GUSARO
10	84	6	M3	10-ene	1	7991
10	72	3	M3	10	2	7991
11	84	6	M3	10	1	7991
11	72	6	M3	10	2	7991
11	70	2	M4	30	2	7991
70	9	1	M4	30	2	912
67	9	1	M4	18	1	912
67	68	1	M2,5	6	2	84
5	1	2	M4	16	3	7991
20	5	2	M4	60	1	912
20	9	1	M4	45	2	7991

47	20	2	M4	35	1	7991
47	80	4	M6	16	1	7984
1	3	1	M4	14	6	7991
2	3	1	M4	14	2	7991
4	3	1	M3	14	2	912
51	52	1	M2,5	10	4	84
72	9	2	M3	16	2	912
94	9	4	M3	14	1	7991
95	9	4	M4	14	1	7991
94	14	4	M3	12	1	912
95	14	4	M4	14	1	912
84	9	2	M3	16	2	7991
84	1	2	M3	16	2	7991
12	84	2	M3	12	2	7991
84	1	2	M3	16	2	7991
73	1	2	M2	12	10	963
86	73	1	M2	6	4	963
87	73	1	M2	6	4	963
87	88	1	M2	8	2	963
1	6	1	M4	14	4	7991
2	6	1	M4	16	2	7991
1	0	1	M4	20	8	7991
1	0	1	M4	14	8	7991
26	6	1	M4	35	4	912
90	26	4	M3	6	4	7991
89	26	4	M3	4	2	7991
13	91	2	M3	6	3	7991
90	38	2	M3	4	1	7991
90	36	1	M3	6	2	7991
17	91	2	M3	6	2	7991
110	111	4	M4	18	1	912
110	111	4	M3	14	1	7991
112	111	4	M3	14	1	912
112	113	4	M3	14	1	912
114	74	2	M3	10	4	7985
107	74	2	M4	12	4	912
109	96	2	M3	16	4	7991
97	73	2	M2	14	4	963
102	73	2	M2	10	8	963
100	104	2	M5	20	2	7991
100	105	2	M5	20	1	7991
101	104	2	M5	20	2	7991
101	105	2	M5	20	1	7991
47	14	2	M4	25	1	912
79	73	2	M2	6	4	84
73	14	1	M2	12	8	963
79	73	1	M2	10	4	84
200	73	1	M2	10	4	84
46	73	1	M2	6	4	963
43	73	1	M2	6	4	963
43	73	1	M2	16	6	963
43	44	12	M4	12	2	7991
44	73	1	M3	12	1	7985
73	22	1	M2	10	10	963
73	24	1	M2	10	10	963
23	22	1	M4	95	1	963

Tabla 7-7 Listado tornillería por unión



# 8. Electrónica de control

Los sistemas mecánicos descritos en capítulos anteriores se encuentran gestionados mediante una electrónica de control. A continuación, se describe la arquitectura hardware relacionada; la tarjeta electronica utilizada y la forma en la que se conectan los diferentes actuadores y sensores de los sistemas mecánicos.

## 8.1. Arquitectura hardware

El Diagrama 8-1 de bloques representa la arquitectura hardware del sistema. En dicha figura se representan como bloques los diferentes actuadores, sensores y elementos electrónicos de control más característicos del sistema. En azul se representan los sensores y actuadores del sistema de stands, en verde los del sistema de palomitas y en gris la electrónica a la que se encuentra conectada. Elementos no relacionados con los sistemas mecánicos desarrollados han sido obviados.

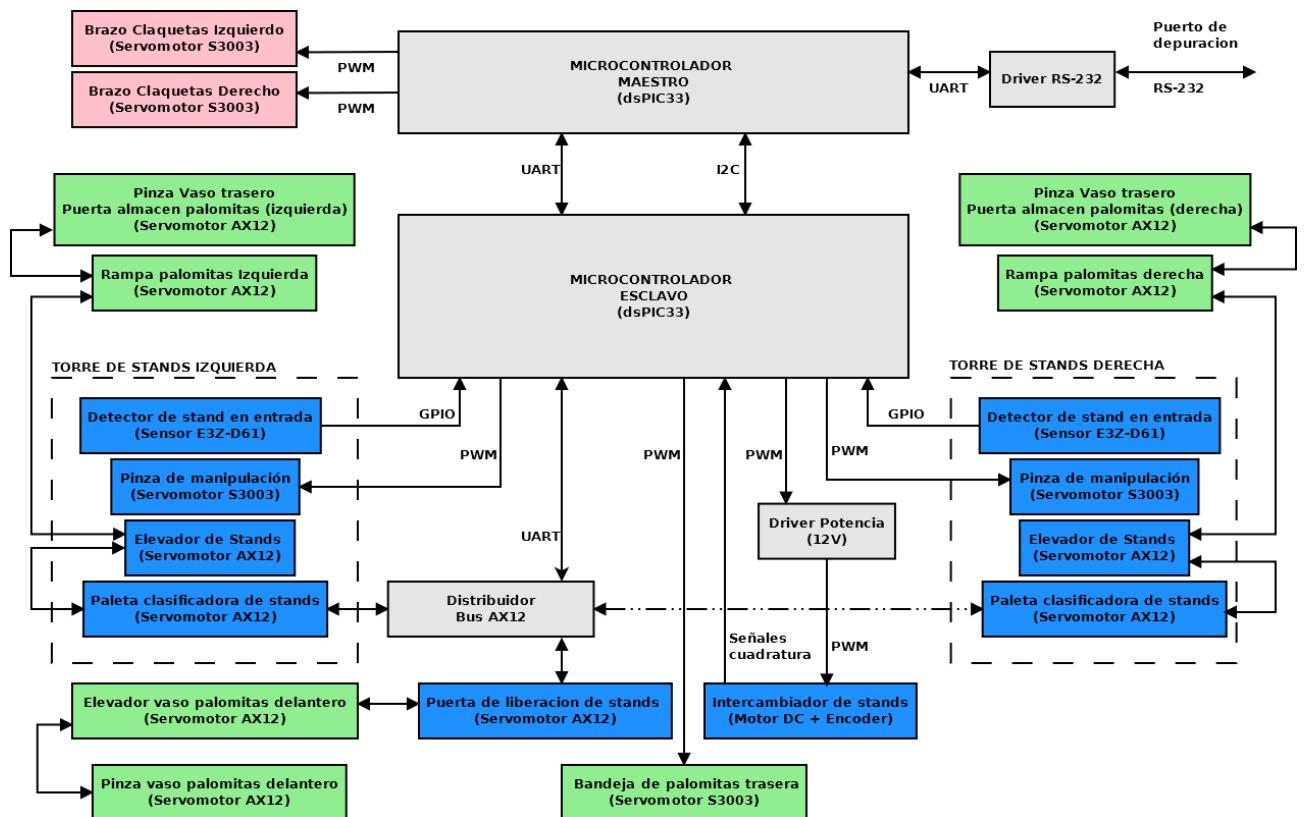


Diagrama 8-1 Arquitectura hardware. Diagrama de bloques.

El sistema se basa en dos microcontroladores dsPIC33 de Microchip: un microcontrolador maestro destinado al control de tracción, evitación de obstáculos y estrategia de juego del robot, y un microcontrolador esclavo destinado al control de los sistemas mecánicos.

Como se observa en el Diagrama 8-1, ambos microcontroladores se encuentran comunicados a través de un bus I2C maestro-esclavo y un bus serie tipo UART full-duplex.

Cada microcontrolador dispone de una interfaz de depuración serie full-duplex, accesible mediante un bus de niveles RS-232. Dicha interfaz se encuentra conectada a uno de los módulos UART del microcontrolador maestro mediante un transceiver RS-232/UART. Realizando un bypass software hacia el segundo módulo UART del dsPIC maestro, es posible acceder a la interfaz de depuración del microcontrolador esclavo.

Por otro lado, la interfaz I2C implementa la interfaz principal de control y monitorización de los sistemas conectados al microcontrolador esclavo.

Según se muestra en el Diagrama 8-1, el microcontrolador maestro controla directamente los actuadores del sistema de claquetas, mientras que el resto de sistemas son gestionados desde el microcontrolador esclavo.

Para la detección de stands dentro del rango de las paletas clasificadoras se han utilizado sensores digitales de luz difusa E3Z-D61 de Omron. Dichos sensores son leídos por el microcontrolador mediante pines tipo GPIO.

Las pinzas de manipulación de stands, la bandeja de palomitas trasera y los brazos utilizados para cerrar claquetas utilizan servomotores o "servos" Futaba S3003. Se trata de servos analógicos que son controlados mediante una señal PWM de 20Hz generada por el microcontrolador a partir de un periférico HW dedicado.

El resto de mecanismos, excepto el intercambiador de stands entre torres, utilizan servos digitales Dynamixel AX12. Estos servos se controlan mediante un bus serie half-duplex que es compartido por todos los servos que estén conectados al bus. Así, para el control de los servos AX12 se utiliza uno de los módulos UART del dsPIC esclavo. Mediante un distribuidor cableado, se generan 3 ramas que interconectan todos los servos AX12.

Por último, el intercambiador de stands utiliza un motor de continua alimentado a 12V mediante un driver de potencia, y controlado mediante una señal PWM de 19Khz. Dicha señal es generada desde el microcontrolador utilizando un módulo HW generador de señales PWM. El motor cuenta con un encoder en cuadratura fijado en su eje trasero, y cuyas señales son decodificadas mediante un módulo HW específico del microcontrolador obteniendo hasta 4 veces el número de pulsos por vuelta de encoder.

## 8.2. Tarjeta de control

La arquitectura HW descrita utiliza la tarjeta Eurobotic2010 diseñada por el equipo Eurobotics Engineering específicamente para aplicaciones orientadas al diseño de robots (ver Figura 8-1).

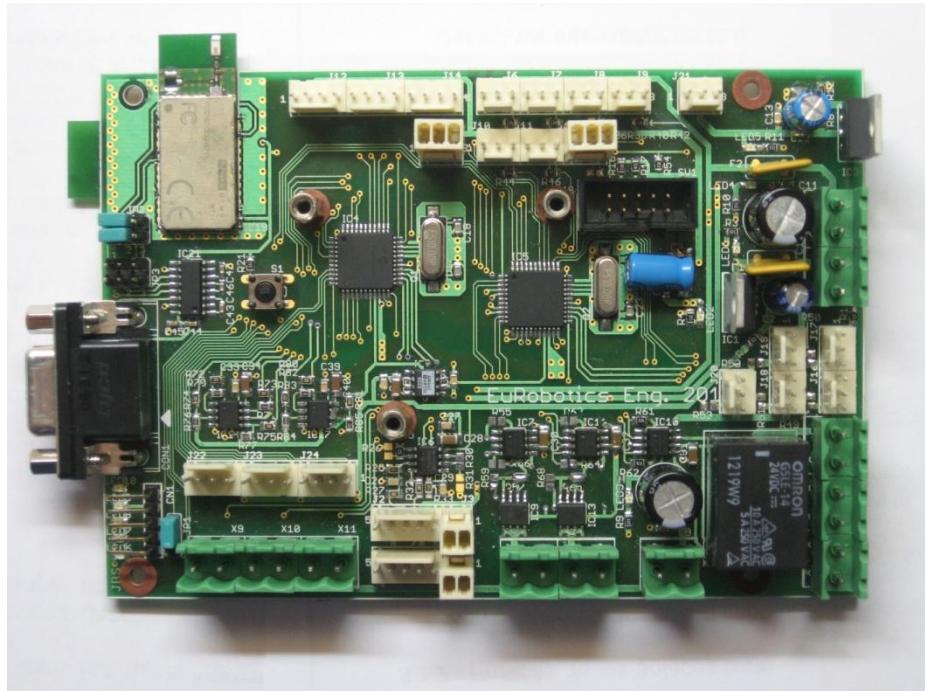


Figura 8-1 Tarjeta electrónica Eurobotics 2010

La tarjeta dispone, como se ha comentado en la sección anterior, de 2 microcontroladores dsPIC33 de hasta 40MIPS. Ambos están comunicados por un bus I2C y una UART full-duplex. Los diferentes recursos de la tarjeta se encuentran repartidos entre los dos microcontroladores, aunque algunos son compartidos pudiendo ser configurados mediante jumpers.

Los recursos del microcontrolador maestro son los siguientes:

- 4 LEDs de depuración.
- 2 entradas analógicas configurables en corriente o tensión (compatible con telémetros láser industriales). Referencia de tensión de 3V discreta.
- 1 entrada digital tipo pull-up (5V).
- 2 entradas de encoders de 2 canales en cuadratura (decodificación por HW).
- 2 puertos para servos analógicos (20Hz, 5V).
- 2 puertos con interfaz compatible con controladoras de motores Dunkermotoren (control de 3 motores).
- Interfaz serie RS-232 compartida.
- 2 interfaces serie Bluetooth. Una compartida y otra dedicada.
- Ampliación de periféricos por bus I2C.

Los recursos del microcontrolador esclavo, destinado al control de la mecánica, son los siguientes:

- 1 LED de depuración.
- 7 entradas para sensores digitales industriales (12-24V) con pull-up o pull-down configurables.
- 1 entradas de encoder de 2 canales en cuadratura (decodificación por HW).
- 3 puertos para servos analógicos (20Hz, 5V).
- 1 puerto para servos AX12 (1Mbps, 12V).
- 2 salidas de potencia en puente en H (12V, 5A).
- 1 salida a relé (2 circuitos)
- 1 puertos con interfaz compatible con controladoras de motores Dunkermotoren (control de 3 motores).
- Interfaz serie RS-232 compartida.
- 1 interfaces serie Bluetooth compartida.
- Ampliación de periféricos por bus I2C.

La tarjeta se alimenta a 5V, 12V y 24V. Los 5V de entrada son utilizados para alimentar servos analógicos y son internamente regulados a 3.3V para alimentar otros dispositivos lógicos. Los 12V son utilizados para alimentar los servos AX12 y los drivers en puente en H destinados al control de motores de continua. Por último, los 24V se utilizan como tensión de alimentación de motores Dunkermotoren, utilizados en la tracción del robot o mecanismos de manipulación de elementos de par elevado.

### 8.3. Sensor E3Z-D61 de OMRON

Sensores E3Z-D61 de OMRON han sido utilizados para la detección de stands a la distancia en la que pueden ser capturados por las paletas, así como la distancia a la que puede ser capturado el vaso trasero por medio de las pinzas.



Figura 8-2 Sensor E3Z-D61 de OMRON

Se trata de un sensor fotoeléctrico de luz infrarroja de reflexión difusa. El modelo utilizado tiene como salida una línea conectada internamente en configuración de colector abierto como se puede ver en la Figura 5.19, con una limitación máxima de entrada de

corriente de 100 mA. Esta configuración permite tanto accionar pequeñas cargas, como adaptar el nivel de tensión en la entrada del dispositivo de control a través de resistencias de pull-up. Tiene dos líneas exclusivas para alimentación con un amplio rango de alimentación soportable.

Las principales características eléctricas son las siguientes:

- Salida: NPN Colector abierto.
- Voltaje residual en colector: 2V.
- Corriente máxima carga: 100 mA max.
- Tensión máxima carga: 26,4 Vdc max.
- Tensión de alimentación 12 a 24Vdc, +/-10%, ripple (p-p): 10% max
- Consumo corriente 30 mA max.
- Respuesta operación o reset: 1ms max.

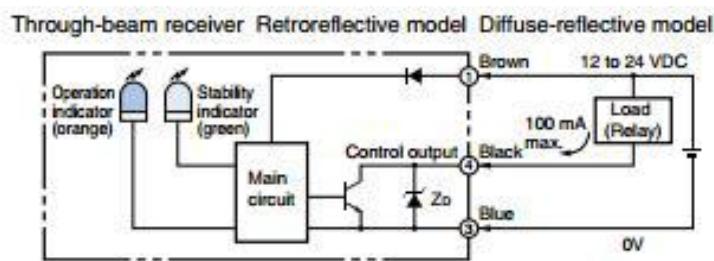


Figura 8-3 Circuito de salida sensor E3Z-D61(Omron, 2005)

El circuito de salida de este sensor se muestra en la Figura 8-3. De acuerdo con este circuito el sensor ha sido conectado a una de las entradas digitales de la tarjeta en configuración pull-up.

## 8.4. Servos Futaba S3003

Para sistemas con exigencia de bajo par como la pinza de manipulación de stands o los brazos utilizados para cerrar claquetas, se ha utilizado un servomotor o “servo” Futaba S3003.

Las principales características eléctricas son las siguientes:

- Alimentación: 4.8 a 6 vcc •10%
- Consumo de corriente (4.8V): 7.2 mA/Inactivo
- Consumo de corriente (6V) : 8 mA/Inactivo
- Amplitud pwm: 3-5 V pico a pico señal cuadrada
- Señal de control compatible con TTL 5V



Figura 8-4 Servomotor Futaba S3003

Como se ha comentado en secciones anteriores, este tipo de servomotores se controlan mediante una señal PWM de 20Hz. Los 5 servomotores de este tipo utilizados en este diseño han sido conectados a los puertos específicos para servos de la tarjeta. Estando gestionados 3 de ellos por el microcontrolador esclavo y los 2 restantes por el dsPIC maestro, como se muestra en el Diagrama 8-1.

## 8.5. Servos Dynamixel AX12

---

Los servos AX12 son unos servos digitales diseñados específicamente para aplicaciones de robótica, que permiten controlar hasta 254 dispositivos a través de una misma línea. Entre todas sus características destacan la lectura de la posición actual y la programación de la velocidad de giro y torque utilizado, disponiendo de un elevado par y velocidad.



Figura 8-5 Servomotor Dynamixel AX12

Las principales características eléctricas de estos servos son las siguientes:

- Alimentación: 9 a 12V (Recomendado 11.1V)
- Consumo de corriente: max 900 mA
- Protocolo comunicación: Half duplex Asynchronous Serial Communication (8bit,1stop,NoParity)
- Señal de control: TTL Level Multi Drop (daisy chain type Connector)

Su protocolo de comunicación es serie asíncrono half-duplex con 8 bits de datos, 1 bit de stop y sin paridad. Siendo la conexión física de niveles TTL Multi Drop. La conexión TTL Multi Drop permite la conexión de muchos dispositivos, a una misma línea de comunicación. Esta línea es conectada a 5V a través de una resistencia de pull-up, y en los dispositivos a una salida configurada en colector abierto. De esta forma el dispositivo que tome el control de la línea enviará los mensajes conectando la línea a masa.

Por otro lado para la conexión de la línea half-duplex de los servos Dynamixel a una UART se recomienda la utilización del circuito de control CM-5 de Dynamixel de la Figura 8-6.

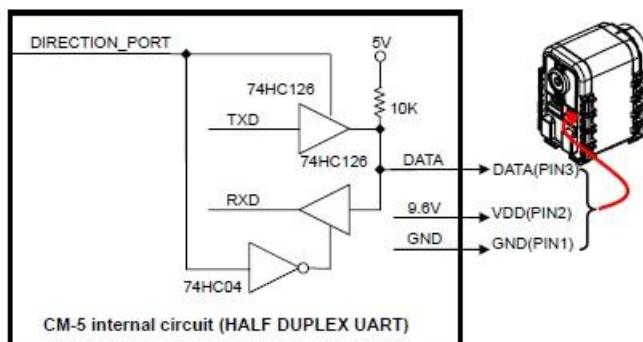


Figura 8-6 Conexión a UART a través de circuito CM-5 de Dynamixel

En la tarjeta utilizada, este circuito se implementa internamente al microcontrolador utilizando los multiplexores internos que permiten conectar los periféricos del microcontrolador a los pines de entrada/salida. Así las líneas TX y RX de la UART utilizadas para controlar los servos AX12, se encuentran conectadas al mismo pin, siendo el protocolo de comunicación el que configura dicho pin como entrada o salida apropiadamente. Al estar conectadas las líneas TX y RX al mismo punto, durante la transmisión de la UART se recibe un eco instantáneo, que es desecharido antes de analizar una trama recibida.

## 8.6. Conjunto Motor y encoder Maxon

El carro horizontal o intercambiador stands se encuentra accionado mediante conjunto 2322.980-52.235-200 Maxon, compuesto por un motor de corriente continua con reductora y un encoder óptico. El microcontrolador esclavo implementa con dicho conjunto un control en lazo cerrado de posición.



Figura 8-7 Conjunto motor, reductora y encoder Maxon.

El encoder, de 100 pulsos por vuelta dispone de dos canales de salida en cuadratura, y necesita de una alimentación de 5 V.

Otras características eléctricas del encoder son las siguientes:

- Frecuencia máxima de operación: 20(KHz)
- Alimentación: 5 Vcc +/- 10%
- Señales de salida compatibles con TTL.
- Salida de corriente por canal min. -1 mA, max. 5 mA

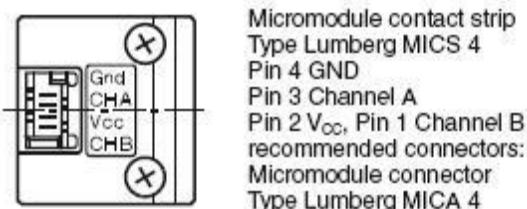


Figura 8-8 Interfaz de conexionado del encoder Maxon ENC20(Maxon)

La Figura 8-8 muestra en planta la interfaz de conexionado del encoder, el cual se encuentra conectado al puerto de entrada de encoder del microcontrolador esclavo. En última medida, se encuentra conectado al módulo decodificador de señales en cuadratura del dsPIC, el cual gestiona de forma HW las cuentas del encoder y realiza un filtrado digital que evita falsas cuentas.

Por otro lado, el motor Maxon tiene una potencia de 6W y una reductora de salida de relación 14:1.

Otras características eléctricas son las siguientes:

- Alimentación nominal: 12 V
- Corriente nominal (max. Continua): 0.807 A
- Velocidad nominal: 4300 rpm
- Revoluciones sin carga: 7010 rpm
- Constante de tiempo mecánica: 13.9 ms
- Resistencia: 5.61 ohm
- Inductancia: 0.492 mH

El motor es controlado mediante una señal PWM de 19KHz que es amplificada hasta la tensión nominal del motor por medio de un driver en puente en H basado en transistores MOSFETs. El control de dicho puente en H se realiza en modo bipolar utilizando un módulo HW dedicado y específico del microcontrolador esclavo.

Por último, dada la baja corriente nominal máxima del motor, de aproximadamente 1A, frente a la corriente máxima del puente en H de 5A (sin tener en cuenta las pérdidas de potencia en las conmutaciones) se consideró que dicho puente en H estaba bien dimensionado. Posteriormente se verificó en la práctica, que los transistores del puente en H eran capaces de disipar la potencia media demandada por el motor en condiciones de trabajo.



## 9. Software de control

---

En última medida, los sistemas mecánicos se encuentran gestionados por un software que permite sincronizar actuadores y sensores con el fin de generar secuencias de movimientos complejas y precisas.

Como se ha visto en el capítulo 8 "Electrónica de control". La mayor parte de los actuadores se encuentran conectados al microcontrolador esclavo o de mecánica. El microcontrolador maestro o de estrategia controla el funcionamiento de estos sistemas mecánicos mediante la interfaz I2C que une a ambos microcontroladores.

En las secciones siguientes se describe funcionalmente el software desarrollado para los microcontroladores maestro y esclavo, relacionado con el control de sistemas mecánicos.

El software desarrollado utiliza o se apoya en las librerías *Open Source* Aversive4dsPIC desarrolladas por el equipo *Eurobotics Engineering*, a partir de las librerías Aversive. Estas últimas, fueron inicialmente desarrolladas por el equipo de Eurobot *Microb Technology* para microcontroladores AVR de Atmel. Posteriormente el equipo *Eurobotics Engineering* añadió la compatibilidad con microcontroladores dsPIC de Microchip.

Este capítulo describe la implementación funcional del software de control desarrollado. Los detalles de la implementación software se pueden consultar en el apéndice de este libro o en el código fuente del CD adjunto al proyecto.

### 9.1. Análisis previo

---

Previo al desarrollo software se ha tratado de clasificar los sistemas mecánicos e identificar patrones que facilitasen el desarrollo software mediante una implementación modular y jerárquica.

Se identifican 3 sistemas que han de funcionar de forma concurrente:

- Sistema de palomitas
- Sistema de stands
- Sistema de claquetas

Esto es, el sistema de palomitas puede estar cogiendo un vaso y al mismo tiempo el sistema de stands almacenando un stand, y el sistema de claquetas preparando uno de los brazos para cerrar una claqueta. En la práctica, esto se ha resuelto, para los dos primeros sistemas, implementando máquinas de estados no bloqueantes que se ejecutan en el bucle principal del programa (*background*). En el caso del sistema de claquetas, este se ejecuta en el microcontrolador maestro, lo que garantiza una ejecución concurrente.

Cada **sistema** está compuesto por **sensores y mecanismos** que combinados y correctamente sincronizados, permiten ejecutar **secuencias** que implementan diferentes **modos de funcionamiento** de cada sistema.

## 9.2. Arquitectura software

El software desarrollado se divide en capas de abstracción, cada capa puede contener varios bloques que utilizan o se apoyan en bloques de la capa inferior. Un bloque que esta a la altura de varias capas indica que tiene acceso a los bloques de cada una de esas capas.

El diagrama de capas del software del microcontrolador esclavo se muestra en la Figura 9-1. Este incluye las capas y bloques principales. Los bloques grises se identifican con módulos utilizados de las librerías Aversive4dsPIC, en azul se representan los bloques desarrollados en este proyecto y en blanco los módulos HW del microcontrolador.

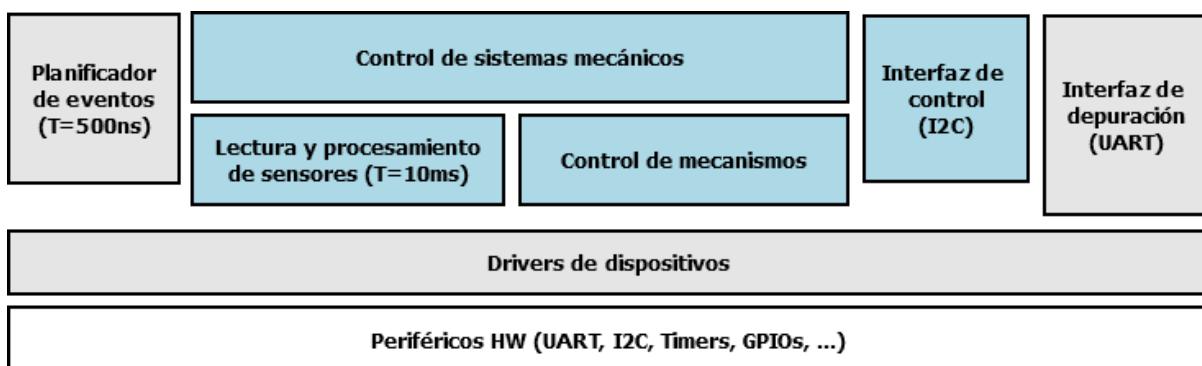


Figura 9-1 Diagrama de capas principal de la arquitectura software.

En la capa más inferior se encuentra el hardware, incluye los periféricos o módulos HW del microcontrolador que forman la interfaz con dispositivos como sensores, actuadores o canales de comunicación.

Por encima de la capa de periféricos HW se encuentra la capa de los drivers que permiten controlar dichos periféricos, realiza una función de abstracción del HW. En este caso, se han utilizado los drivers proporcionados por la librería Aversive4dsPIC.

El resto de capas superiores implementan el nivel de aplicación, no dependiente del HW. Esto es, con los drivers apropiados para acceder al HW, estas capas podrían correr en cualquier otra máquina.

La capa de aplicación está compuesta por los siguientes bloques:

- Planificador de eventos
- Interfaz de depuración
- Control de sistemas mecánicos
- Interfaz de control

El **planificador de eventos**, utiliza el módulo *scheduler*, es un módulo de las librerías Aversive4dsPIC que permite ejecutar eventos de forma síncrona o periódica. Cada evento tiene asociado una función, un periodo y una prioridad. El scheduler tiene asociado un periodo base de ejecución. En cada periodo de ejecución el scheduler determina qué evento es el que tiene que ejecutarse en función de su prioridad y periodo. El scheduler utiliza un Timer HW y ejecuta los eventos dentro de un contexto de interrupción. El microcontrolador dsPIC permite anidar interrupciones, por lo que un evento de prioridad baja puede ser enmascarado por uno de mayor prioridad. Aun así, las funciones ejecutadas en el *scheduler* deben de ser no bloqueantes y corto tiempo de ejecución para favorecer la ejecutabilidad del sistema.

La **interfaz de depuración** utiliza el módulo *parser* de las librerías Aversive4dsPIC para implementar una línea de comandos serie a través del módulo UART. Cada comando, principalmente, tiene asociado una serie de parámetros de entrada y una función. Cada parámetro de entrada tiene asociado un tipo (*string, integer o float*) y un tamaño (8, 16, 32 bits). La función asociada a un comando sólo se ejecutará si los parámetros de entrada cumplen con sus características y posibles valores. Este módulo, se apoya a su vez en otros módulos de la misma librería hasta llegar al módulo UART. La ejecución de la interfaz de depuración se realiza en el bucle principal del programa o *background*.

El bloque de **control de sistemas mecánicos** gestiona los diferentes modos de funcionamiento de los sistemas de stands y palomitas. Para ello, se apoya sobre el bloque de **lectura y procesamiento de sensores** y el de **control de mecanismos**. Estos tres bloques son accesibles y controlables a través de una **interfaz de control I2C**, utilizada por el microcontrolador maestro para sincronizar estos sistemas mecánicos con movimientos del robot. En las secciones siguientes se describen en detalle cada uno de estos bloques.

### **9.3. Lectura y procesamiento de sensores**

---

Únicamente se dispone de sensores digitales asociados a los sistemas mecánicos, como son los sensores OMRON utilizados para detectar stands dentro del rango de manipulación. La interfaz de este módulo permite acceder al valor actual de cada sensor por su nombre (por ejemplo: S\_STAND\_LEFT) abstrayendo así cualquier dependencia con el hardware.

La lectura de estos sensores se realiza directamente por medio de los puertos de entrada del microcontrolador. Con el fin de evitar falsas lecturas, los valores leídos son previamente procesados. Todo ello se ejecuta periódicamente cada 10ms en un evento del *scheduler*.

La lectura cada 10ms permite filtrar posibles rebotes de alta frecuencia de la señal de los sensores. Además, cada sensor tiene asociado un comparador de nivel con histeresis configurable. Este comparador de nivel permite definir el número de lecturas de un sensor para considerar que se encuentra en un estado de activado o desactivado. Cada umbral de dichos estados es configurable, de esta forma es posible configurar la sensibilidad de activación del sensor y filtrar rebotes de baja frecuencia.

## **9.4. Control de mecanismos**

---

Cada sistema se compone de los siguientes mecanismos:

### **Sistema de palomitas:**

- Rampas de recolección
- Bandeja trasera
- Puerta almacén palomitas
- Pinza vaso trasera
- Pinza vaso delantera

### **Sistema de stands:**

- Palas clasificadoras
- Elevador de stands
- Pinza stands
- Compuertas de torres
- Intercambiador de stands

### **Sistema de claquetas:**

- Brazo izquierdo y derecho

Cada mecanismo puede estar formado por un actuador simple o varios actuadores o sensores, y representa un elemento funcional. Por ejemplo, el intercambiador de stands está compuesto por un motor de continua y un encoder.

El objetivo de esta capa es proporcionar una interfaz de control que abstraiga a las capas superiores el control de actuadores discretos, y permita un control o una gestión a nivel de mecanismos desatendida.

### **9.4.1. Interfaz y funciones de control de mecanismos**

---

Los controladores de mecanismos desarrollados tienen una misma interfaz y métodos de control común. Disponen de los siguientes tipos de funciones:

- Establecimiento de consigna
- Comprobación de consigna alcanzada

La función de **establecimiento de consigna** inicia una trayectoria del actuador hacia el valor de dicha consigna. La consigna puede especificarse en unidades de ángulo, distancia o posiciones discretas previamente configuradas.

Mediante la función de **comprobación de consigna alcanzada** es posible comprobar si una trayectoria a finalizado y de qué forma. La función permite comprobar si una trayectoria ha finalizado de las siguientes formas:

- Posición de consigna alcanzada

- Posición cercana a la consigna
- Bloqueo al intentar alcanzar la consigna

La condición de posición cercana a la consigna permite encadenar varios movimiento o trayectorias de forma suave y en menor tiempo que si se esperase a alcanzar la consigna. La ventana que define la posición cercana a la consigna es configurable.

Por otro lado, la detección de un bloqueo permite actuar consecuentemente para realizar correcciones que posibiliten la acción a realizar o abortarla.

#### 9.4.2. Mecanismos basados en servos analógicos

El diagrama de capas de la implementación de los actuadores basados en servos analógicos se muestra en la Figura 9-2. Cada controlador de mecanismo utiliza el driver proporcionado por la librería Aversive4dsPIC que utilizar módulos de tipos *Output Compare* del microcontrolador para generar la señal PWM necesaria para fijar la posición de los servos.

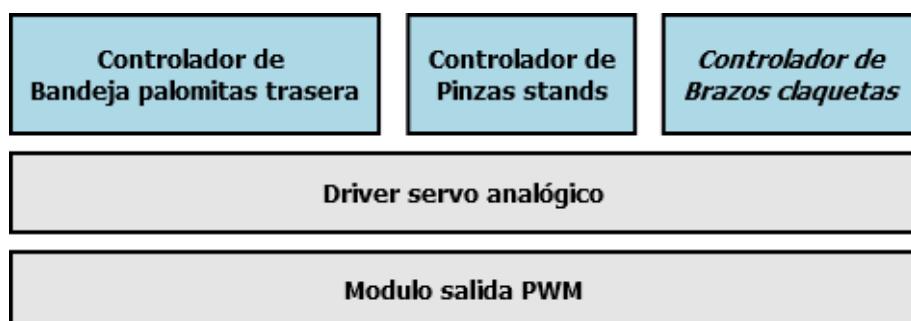


Figura 9-2 Diagrama de capas software del control de mecanismos basados en servos analógicos.

Este tipo de servos no disponen de ningún sensor que permita realimentar su posición o velocidad actual. Es por ello que la detección del fin de trayectoria se realiza de forma temporizada, no pudiendo detectar situaciones de bloqueo.

Notar que aunque los brazos utilizados para cerrar las claquetas no son controlados por el microcontrolador esclavo, la implementación sobre el microcontrolador maestro es idéntica y utiliza las mismas capas inferiores.

#### 9.4.3. Mecanismos basados en servos AX12

La mayoría de los mecanismos utilizan servos AX12. La Figura 9-3 muestra el diagrama de capas (horizontal) de la implementación de los módulos controladores de mecanismos de este tipo.

Los controladores implementados soportan todas las funcionalidades de la interfaz de control descrita anteriormente. Los servos AX12 permiten especificar, además de la posición consigna, parámetros como la velocidad de giro y el par, que pueden ser leídos en todo momento.

Dada la cantidad de mecanismos que utilizan servos AX12 se ha implementado una capa de **control de trayectorias** genérica sobre la que se ha implementado los diferentes controladores.

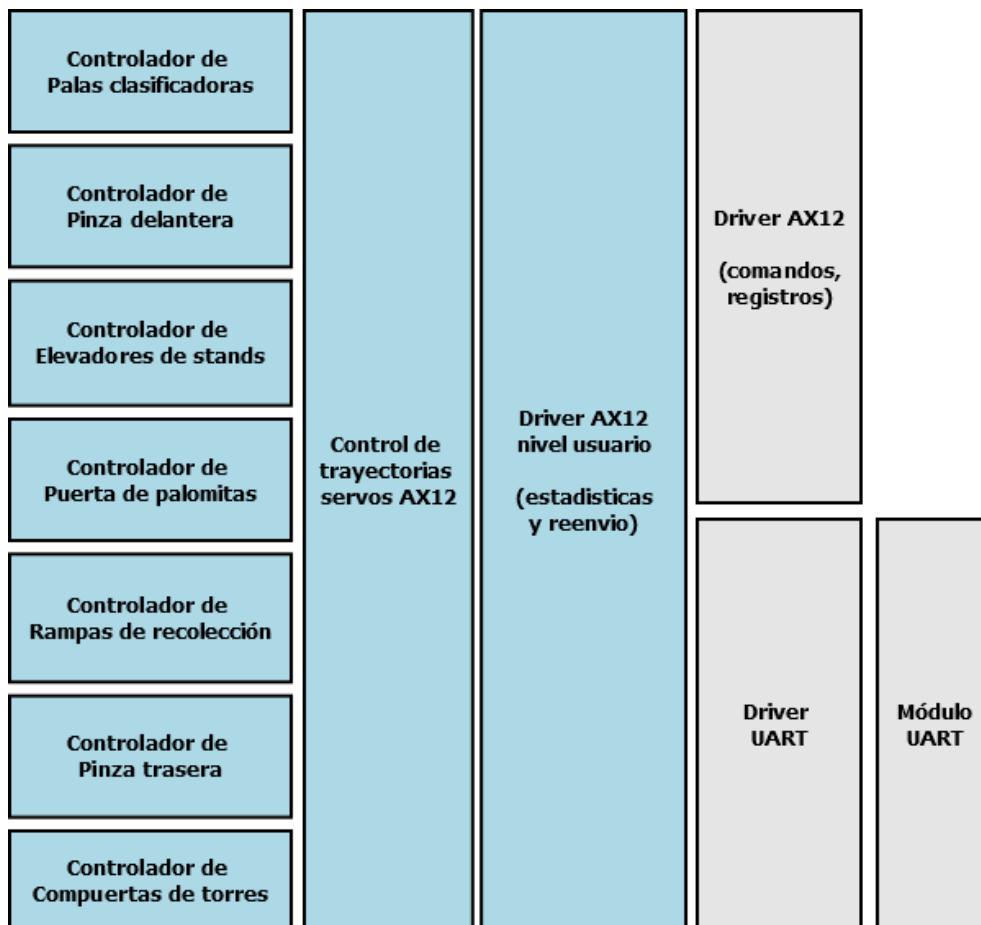


Figura 9-3 Diagrama de capas software del control de mecanismos basados en servos AX12.

Por debajo del control de trayectorias se encuentra el **driver AX12 de usuario**, el cual utiliza el driver AX12 y del módulo UART de las librerías Aversive4dsPIC para controlar los servos AX12. Además, el driver de usuario añade la funcionalidad de reenvío de comandos, y el registro de estadísticas de errores de la comunicación.

#### 9.4.4. Intercambiador de stand

El intercambiador al igual que otros mecanismos necesita ser controlado en posición. Pero a diferencia de los mecanismos anteriores, no utiliza un servomotor que realice el control de dicha posición. El mecanismo intercambiador de stands utiliza un motor discreto cuya posición es medida por un encoder, de forma que el control en posición ha de implementarse externamente. En este caso dicho control se encuentra implementado en el microcontrolador esclavo.

El diagrama de capas del intercambiador de stands se muestra en la Figura 9-4. El módulo controlador de trayectorias constituye la interfaz del mecanismo e implementa las funcionalidades descritas en la sección 9.4.1.

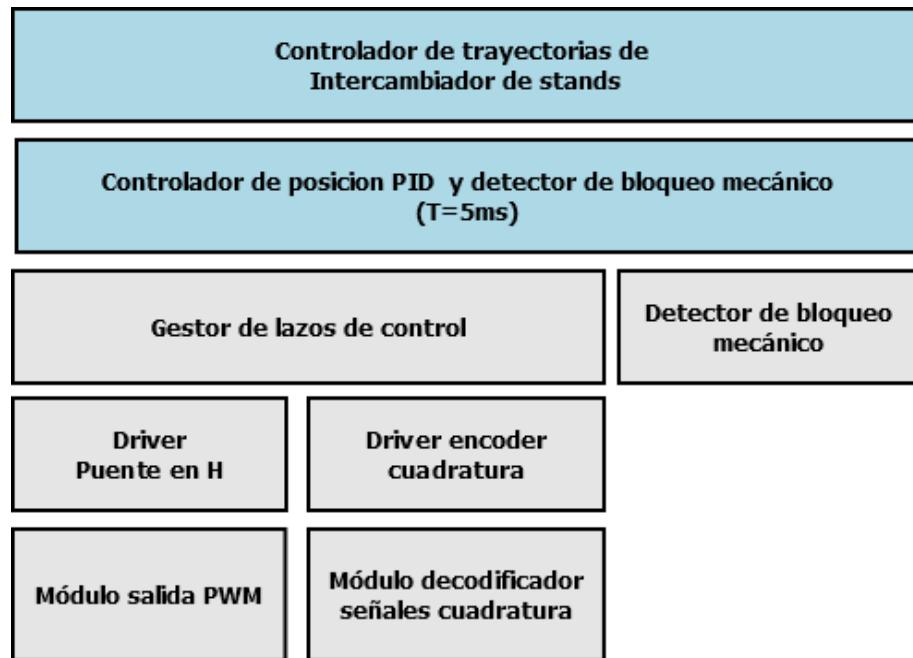


Figura 9-4 Diagrama de capas software del control del intercambiador de stands.

El **control en posición** del intercambiador de stands se implementa mediante un control en lazo cerrado de tipo PID. Dicho lazo de control se ejecuta como una tarea del *scheduler* con un periodo de 5ms. Para la implementación de dicho controlador PID se ha utilizado un módulo de gestión de lazos de control de las librerías Aversive4dsPIC.

La **detección de bloqueo** mecánico se ha implementado mediante la estimación de la corriente del motor. Este método utiliza la tensión a la que se está alimentando el motor y la velocidad instantánea para calcular un valor estimado de la corriente instantánea. La **condición de bloqueo** se producirá si la velocidad del motor está por debajo de un **umbral de velocidad** y la corriente instantánea supera un **umbral de corriente** durante un **tiempo mínimo**.

La detección de bloqueo ha sido implementado utilizando un módulo de las librerías Aversive4dsPIC. Implementa el método descrito anteriormente. Este módulo obtiene la tensión y velocidad instantánea del motor, necesarias para la estimación de la corriente, del gestor de lazos de control.

## 9.5. Gestión de mecanismos y sistemas mecánicos

---

La gestión de la mecánica se ha dividido en dos niveles atendiendo a su nivel de abstracción:

- Gestión de mecanismos
- Gestión de sistemas mecánicos

Ambos niveles son accesibles por medio de la interfaz de control I2C y se ejecutan en el bucle principal del programa.

La **gestión de mecanismos** permite realizar operaciones a nivel de mecanismos:

1. Recibe la consigna de posición a través de la interfaz I2C.
2. Establece la consigna del mecanismo a controlar.
3. Espera que la trayectoria hacia la nueva consigna haya terminado.
4. Notifica a través de la interfaz I2C la forma en la que ha terminado la trayectoria.

Este nivel está implementado como una función lineal, que debido al punto 3 es bloqueante. Es por ello que en la práctica este tipo de gestión de mecanismos no se utiliza. Este nivel fue implementado en una primera fase, con el objeto de diseñar, probar y calibrar paso a paso las secuencias de mecanismos que luego han sido implementadas en la gestión de los sistemas mecánicos.

La **gestión de sistemas** permite realizar operaciones a nivel de sistemas mecánicos completos. Como se ha comentado anteriormente, un sistema permite ejecutar un **modo de operación** que a su vez ejecuta **secuencias** de mecanismos sincronizadas con sensores.

Este nivel tiene una implementación tipo máquina de estados que permite la ejecución concurrente de los diferentes sistemas:

- Sistema de palomitas
- Sistema de torre de stands izquierda
- Sistema de torre de stands derecha

La gestión de cada sistema funciona siguiendo el siguiente flujo:

1. Petición por medio de la interfaz de control I2C de un modo de funcionamiento.
2. Ejecución una o varias secuencias de mecanismos.
3. Notificación por medio de la interfaz de control I2C de la finalización del modo.

Durante las secuencias de mecanismos se encadenan trayectorias sincronizadas con otros mecanismos o sensores. Durante la ejecución de un modo de funcionamiento el sistema notifica en todo momento su estado por medio de la interfaz de control I2C. De esta forma, el microcontrolador maestro puede determinar si un modo de funcionamiento se encuentra en una fase en la que puede iniciar el movimiento del robot, o de si el modo se ha ejecutado satisfactoriamente o no.

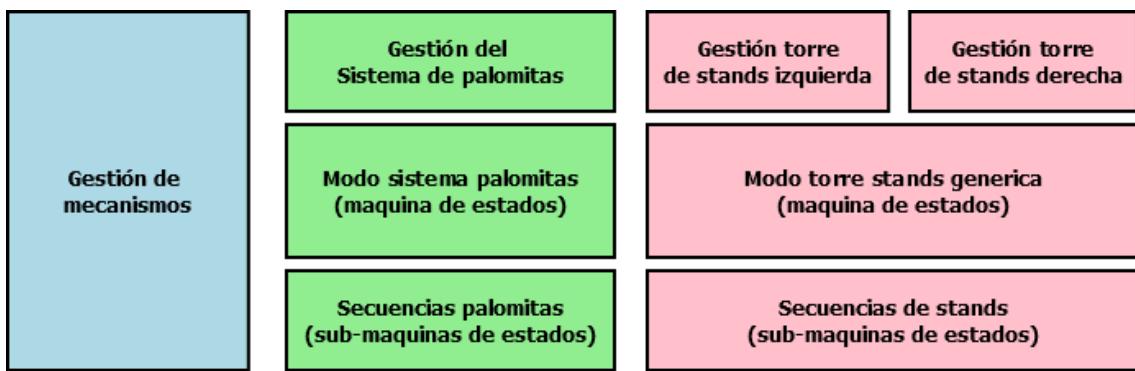


Figura 9-5 Diagrama de capas software de la gestión de mecanismos y sistemas mecánicos

El funcionamiento descrito anteriormente se ve reflejado en el diagrama de capas de la Figura 9-5. Notar que en el caso del sistema de palomitas, dada su simetría en sensores y actuadores, se ha desarrollado un bloque genérico de gestión de una torre que tiene asociada una estructura de datos que representa a los actuadores y sensores de dicha torre. Una estructura de datos asociada a cada torre permite reutilizar el mismo bloque para implementar la gestión de ambas.

## 9.6. Interfaz de control I2C

La interfaz de control I2C permite que el microcontrolador maestro gestione los sistemas mecánicos controlados por el microcontrolador esclavo. La interfaz tiene asociado un protocolo de comunicación de comandos de lectura y escritura. Dada la configuración del bus I2C maestro-esclavo es el microcontrolador maestro el que siempre realiza las peticiones de lectura o escritura.

Los **comandos de escritura** se corresponden con **comandos de cambio de modo de funcionamiento** de un sistema o mecanismo. Los **comandos de lectura** son **peticiones de estado**. En dichas peticiones, el microcontrolador esclavo comparte información de estado de los diferentes mecanismos, sistemas o sensores. El microcontrolador maestro realiza la **petición de estado de forma periódica**, mientras que los comandos de cambio de modo suceden de forma aleatoria dependiendo del comportamiento del robot.

El diagrama de capas de la Figura 9-6 muestra los principales bloques funcionales de la interfaz de control I2C del microcontrolador esclavo. Los comandos de escritura y lectura son procesados como eventos cada vez que el driver de bus I2C recibe una trama completa de escritura o de lectura.

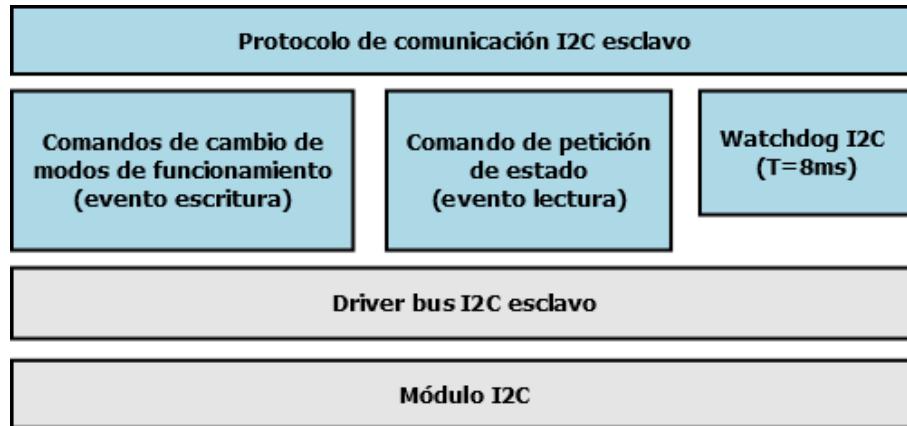


Figura 9-6 Diagrama de capas software de la interfaz de control (microcontrolador esclavo)

En paralelo a la recepción de comandos, existe un *watchdog* que comprueba que la petición de estado periódica se realiza. En caso contrario, reinicia el protocolo y el bus I2C con el fin de restablecer un posible fallo del bus. Este *watchdog* se ejecuta cada 8ms en un evento del *scheduler*.

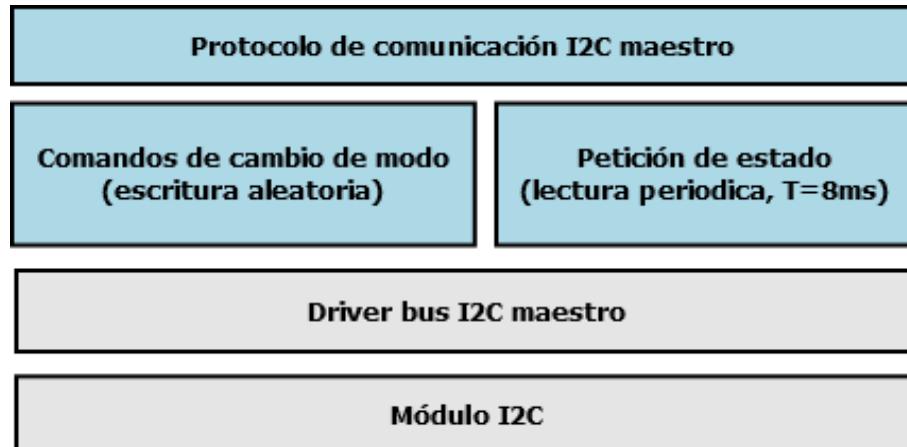


Figura 9-7 Diagrama de capas software del interfaz de control (microcontrolador maestro)

Por otro lado, en el microcontrolador maestro se encuentra la parte complementaria y simétrica del protocolo de comunicación. En este caso, un evento periódico realiza **peticiones de estado cada 8ms**, y según las necesidades de comportamiento del robot envía **comandos de cambio de modo** para gestionar los mecanismos o sistemas mecánicos.

# PLANOS

Todos los planos y sus acotados han sido dibujados siguiendo la Norma de estandarización internacional ISO 129-1:2004 (International Organization for Standardization, N°129, apartado 1).

Los planos han sido creados incluyendo todas las informaciones dimensionales necesarias para definir completamente cada una de las piezas. Siendo esta una exigencia de la norma la realidad es que actualmente con los sistemas de fabricación por control numérico y el requerimiento para la fabricación del soporte CAD de la pieza diseñada, las características del acotado tienden a centrarse en dar soporte a las posteriores labores de comprobación del mecanizado. En cualquier caso los planos han sido creados definiendo completamente cada una de las piezas diseñadas.

Se ha diseñado un cajetín para incluir toda la información necesaria para la fabricación. En este cajetín se ha incluido la siguiente información:

- TITULO: Nombre de la pieza
- NÚMERO PLANO: Para este campo se ha creado un sistema de codificación del número de identificación de los planos que va a ser explicado a continuación.
- HOJA: Formato estandarizado de hoja en el que ha sido creado el plano y en el que deberá ser impreso.
- ESCALA: Escala principal en la que se muestra el dibujo. Pueden existir detalles de partes de piezas representados a distinta escala. En estos casos se indica la escala bajo la vista de detalle.
- "CAMPO" HOJA X DE Y: Y indica el número de hojas en las que se han incluido todas las vistas de acotado de la pieza. X identifica la hoja dentro del conjunto Y.
- NOMBRE: Siglas de la persona que ha creado el dibujo.
- FECHA: Fecha de creación del dibujo.
- MATERIAL: Material de fabricación.
- ESPESOR: Espesor del material. En algunas piezas de corte en dos dimensiones en las que no se incluye la vista de alzado, este dato define el espesor de la pieza terminada.
- UNIDAD: Número de unidades por conjunto. En este campo se puede indicar si se deben de fabricar piezas simétricas. En este caso se indica mediante las siglas N°(S/D) + N°(M/C). Siendo "S/D" Segundo/Dibujo y "M/C" Mano/Contraria.
- "CAMPO" Unidades y tolerancias generales. En este campo se indican las unidades de las cotas y las tolerancias que deberán ser utilizadas para todas aquellas cotas que no dispongan de tolerancia específica o notación de ajuste.

Para todos los planos se han definido la unidad de acotado en milímetros, siendo siempre el valor de cota la medida de la magnitud real de pieza, con independencia de la escala a la que este dibujado el plano.

Para la numeración de los planos se ha creado un sistema de codificación por campos, para facilitar la rápida identificación del plano entre los diferentes conjuntos del sistema sus subconjuntos y cada una de las piezas de estos.

El formato es el siguiente:

XX-YY-ZZ-"(LETRA) o N" NN

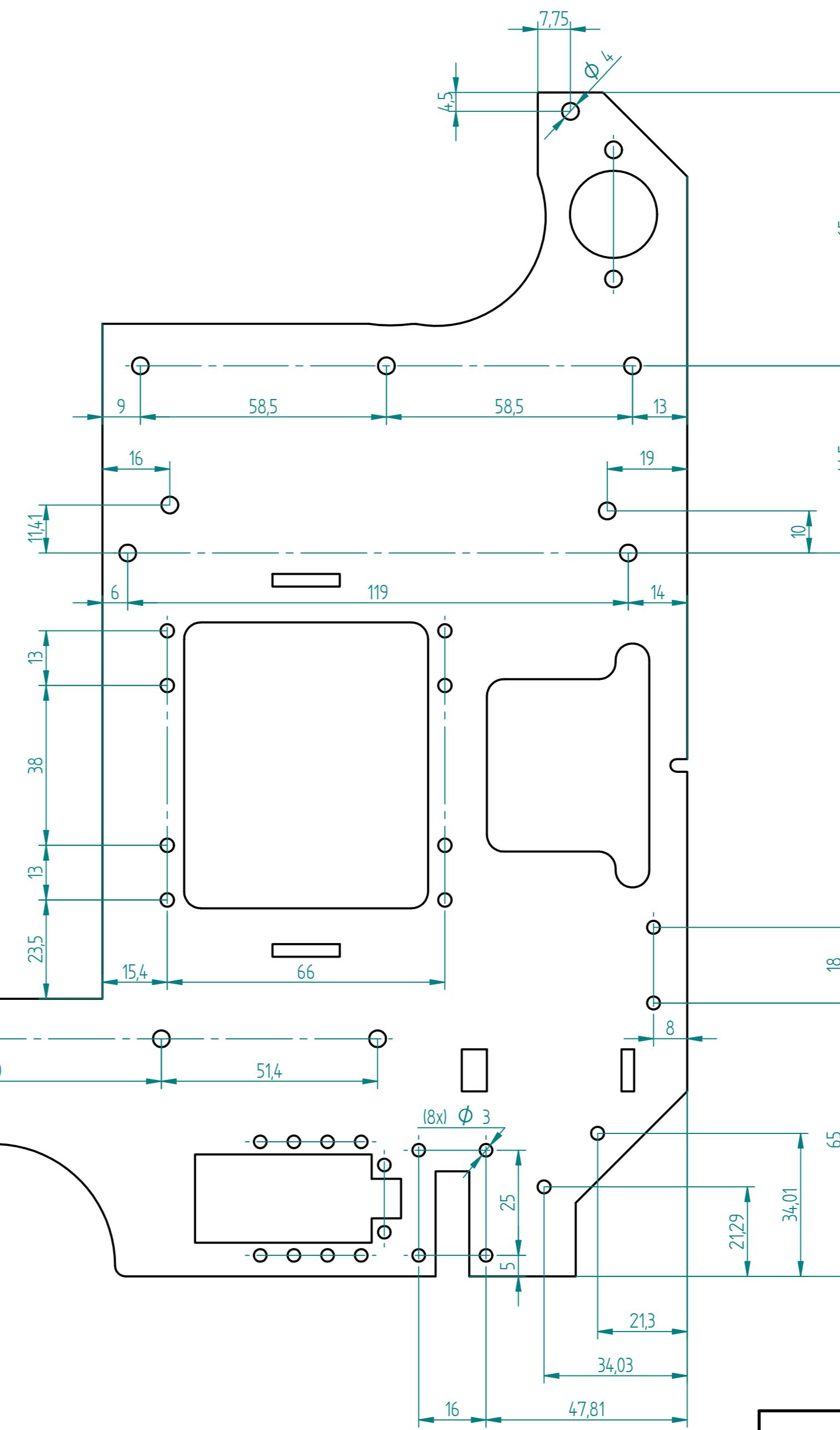
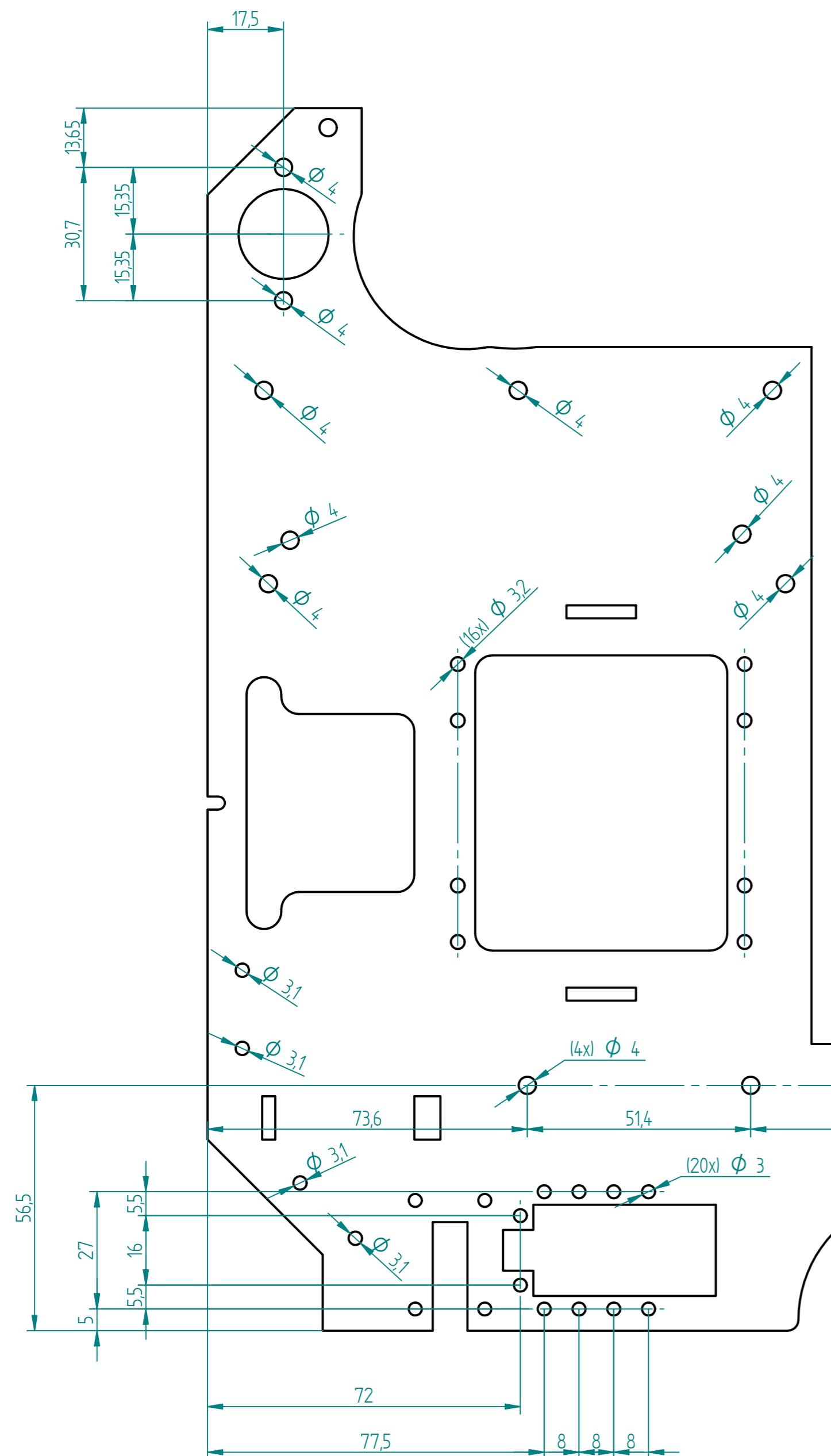
- En el campo XX se han codificado cada uno de los sistemas o conjuntos principales que forman el diseño. Estos son:
  - 00 ROBOT
  - 01 BLOQUE MOTOR
  - 02 LASTRE
  - 03 BALANCÍN ENCODER
  - 04 CONJUNTO CARRO 2
  - 05 PALETAS STAND
  - 06 CARRO HORIZONTAL
  - 07 PUERTA STAND
  - 08 PINZA VASO DELANTERO
  - 09 BRAZO CLAQUETA
  - 10 BRAZOS PALOMITAS
  - 11 PINZA VASO
  - 12 TRAMPILLA SUPERIOR
  - 14 ELECTRÓNICA

En el bloque motor se incluyen todas las piezas de la estructura y aquellas que no estén localizadas en alguno del resto de subconjuntos creados.

- En el campo YY se codifican cada uno de los subconjuntos que componen un conjunto principal. El conjunto "00" siempre está integrado por las piezas que no pertenecen a ningún subconjunto.
- En el campo ZZ se codifica el número de cada una de las piezas que componen un subconjunto.
- El campo NNN es un código único asignado a cada una de las piezas de forma correlativa a su creación y que ha sido utilizado para las operaciones de montaje y creación de listados como el de tornillería por unión de cada una de las piezas.
- Por último el campo (LETRA) ha sido creado para identificar planos con características especiales.

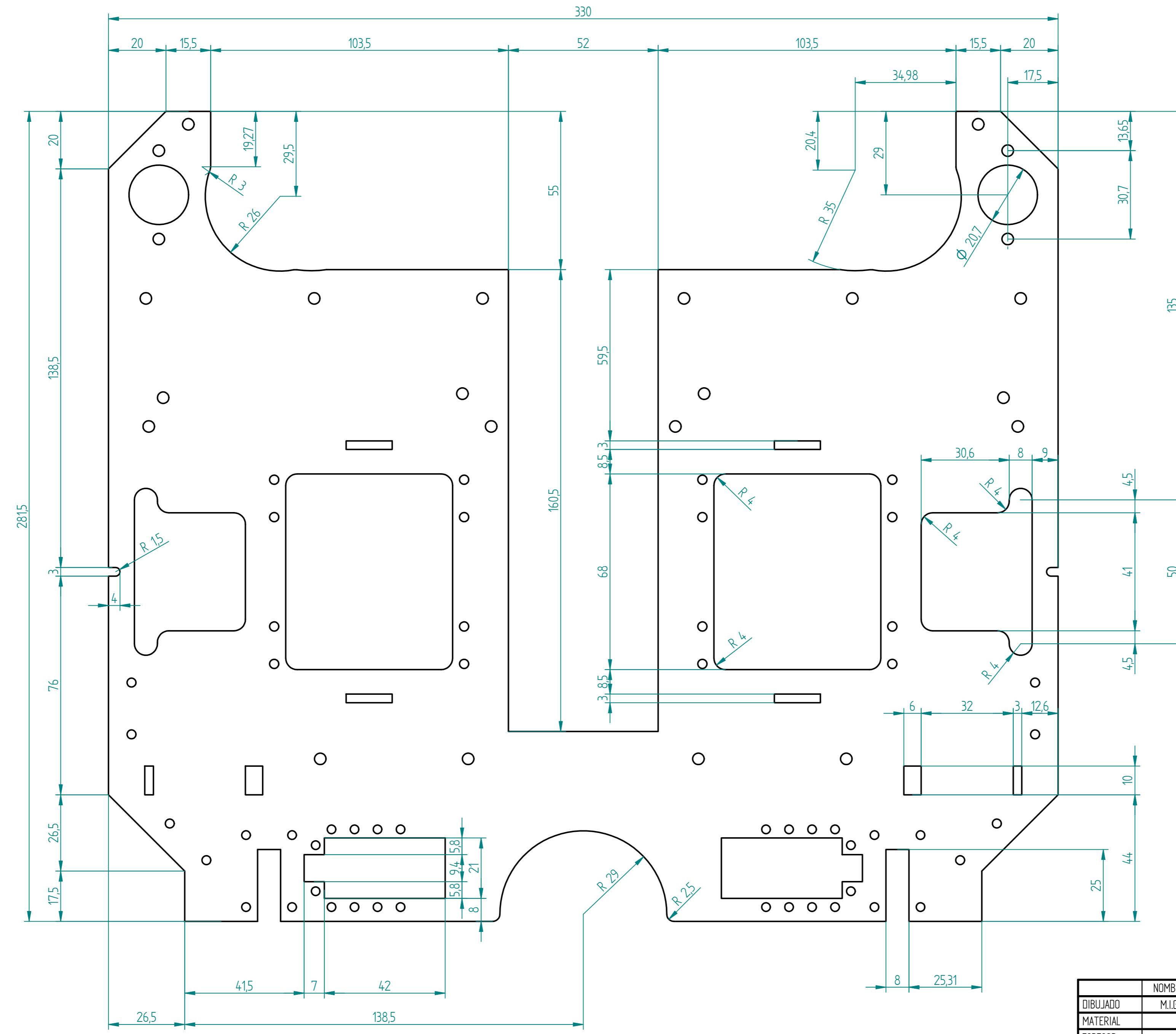
Las letras definidas son las siguientes:

- E(NN) Número de explosión de conjunto.
- C(NN) Número de plano de conjunto de detalle de sistema.



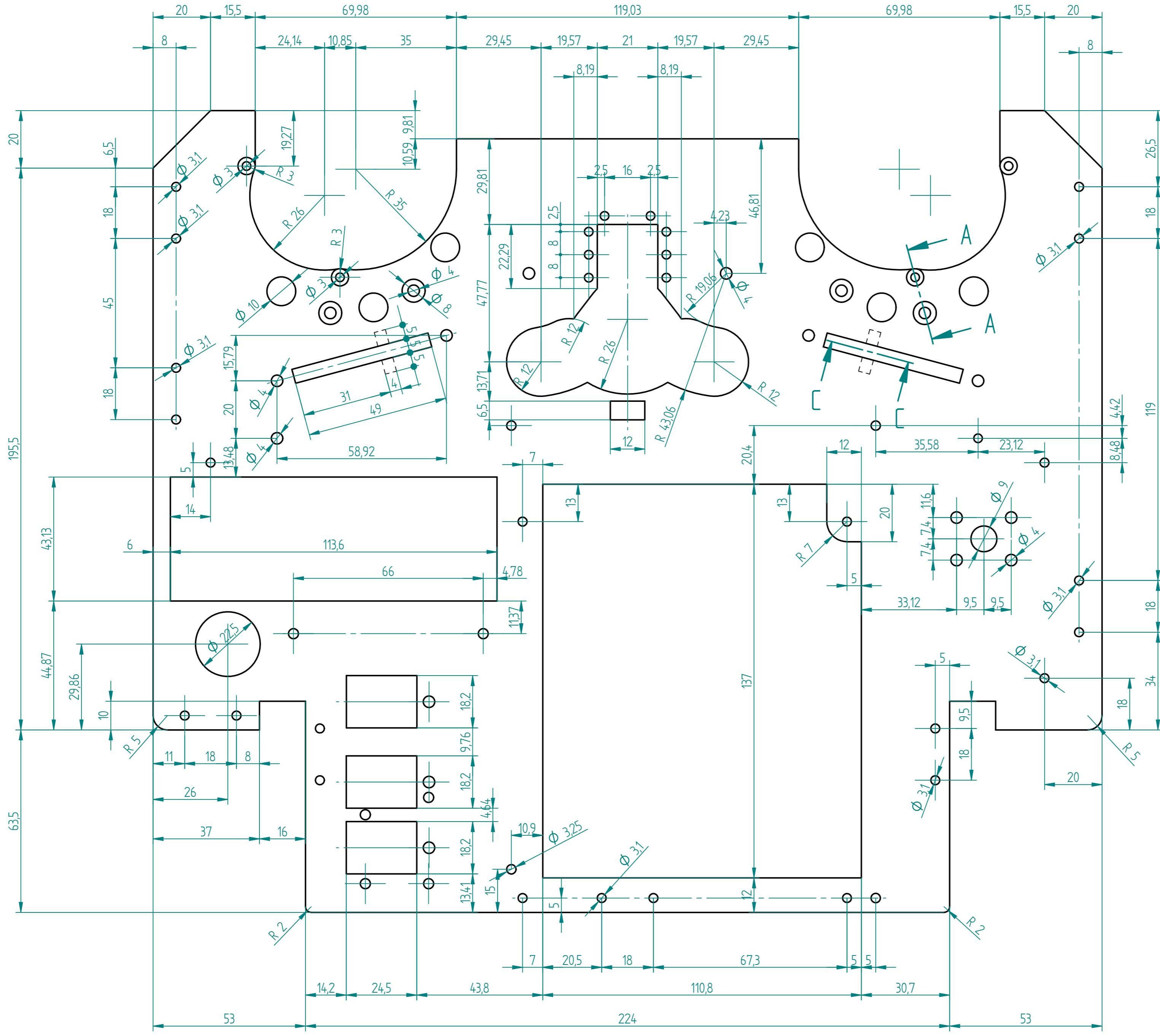
	NOMBRE	FECHA	Proyecto Eurobot 2015		
DIBUJADO	M.I.G	05-07-15			
MATERIAL	TRESPA		TITULO BASE 1		
ESPESOR	6 MM				
CANTIDAD	1 UNIDADES				
DIMENSIONES ACOTADAS EN MILIMETROS TOLERANCIA GENERAL $\pm 0.2$ ANGULO $\pm 1^\circ$			HOJA A2	NUMERO PLANO 00-01-01-001	REV 0
			ESCALA: 1:2	PESO:	HOJA 1 DE 2





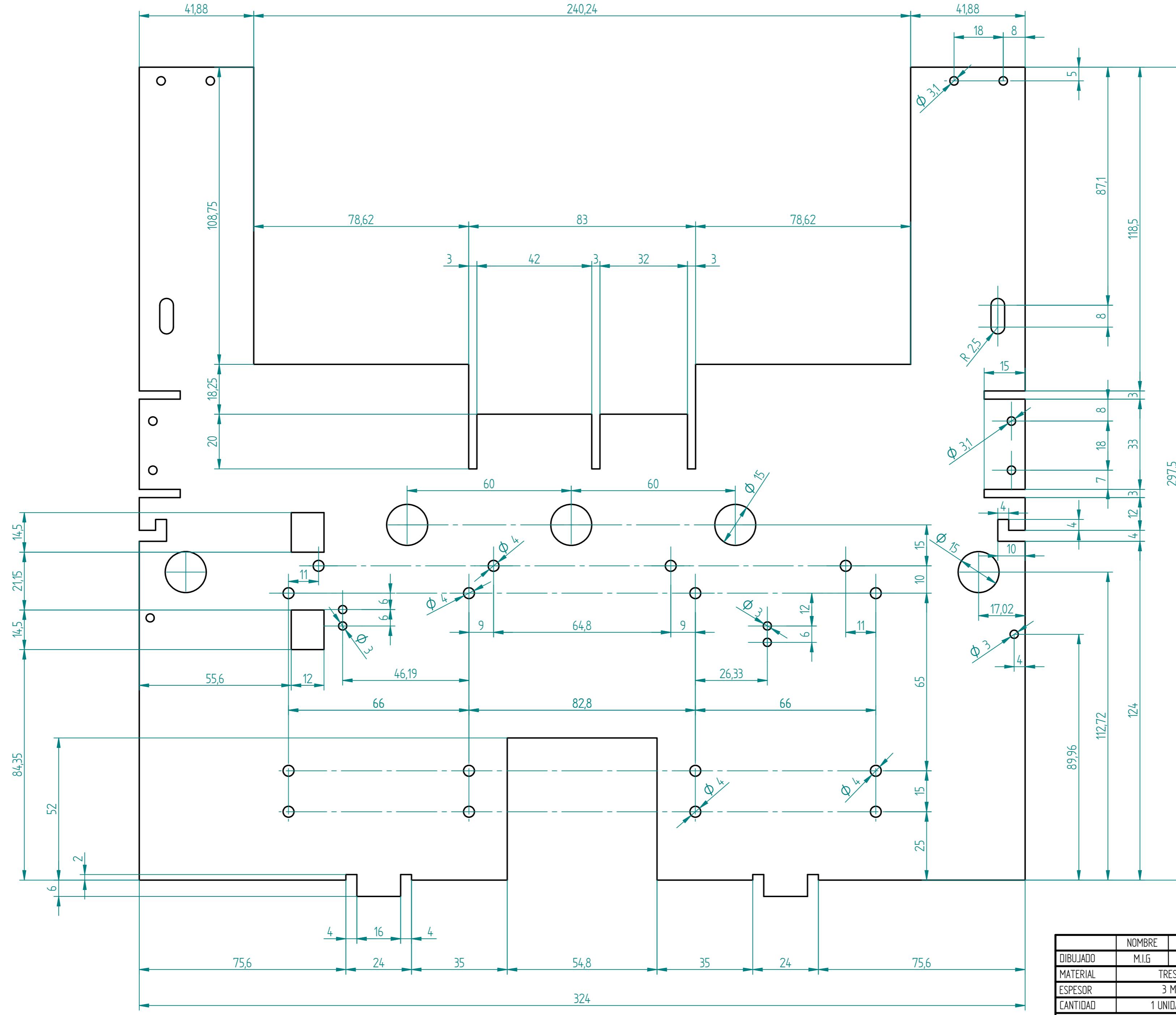
	NOMBRE	FECHA	Proyecto Eurobot 2015		
DIBUJADO	M.I.G	05-07-15			
MATERIAL	TRESPA	TITULO			
ESPESOR	6 MM	BASE 1			
CANTIDAD	1 UNIDADES				
DIMENSIONES ACOTADAS EN MILIMETROS			HOJA	NUMERO PLANO	REV
TOLERANCIA GENERAL $\pm 0.2$			A2	00-01-01-001	0
ANGULO $\pm 1^\circ$			ESCALA: 1:2	PESO:	HOJA 2 DE 2





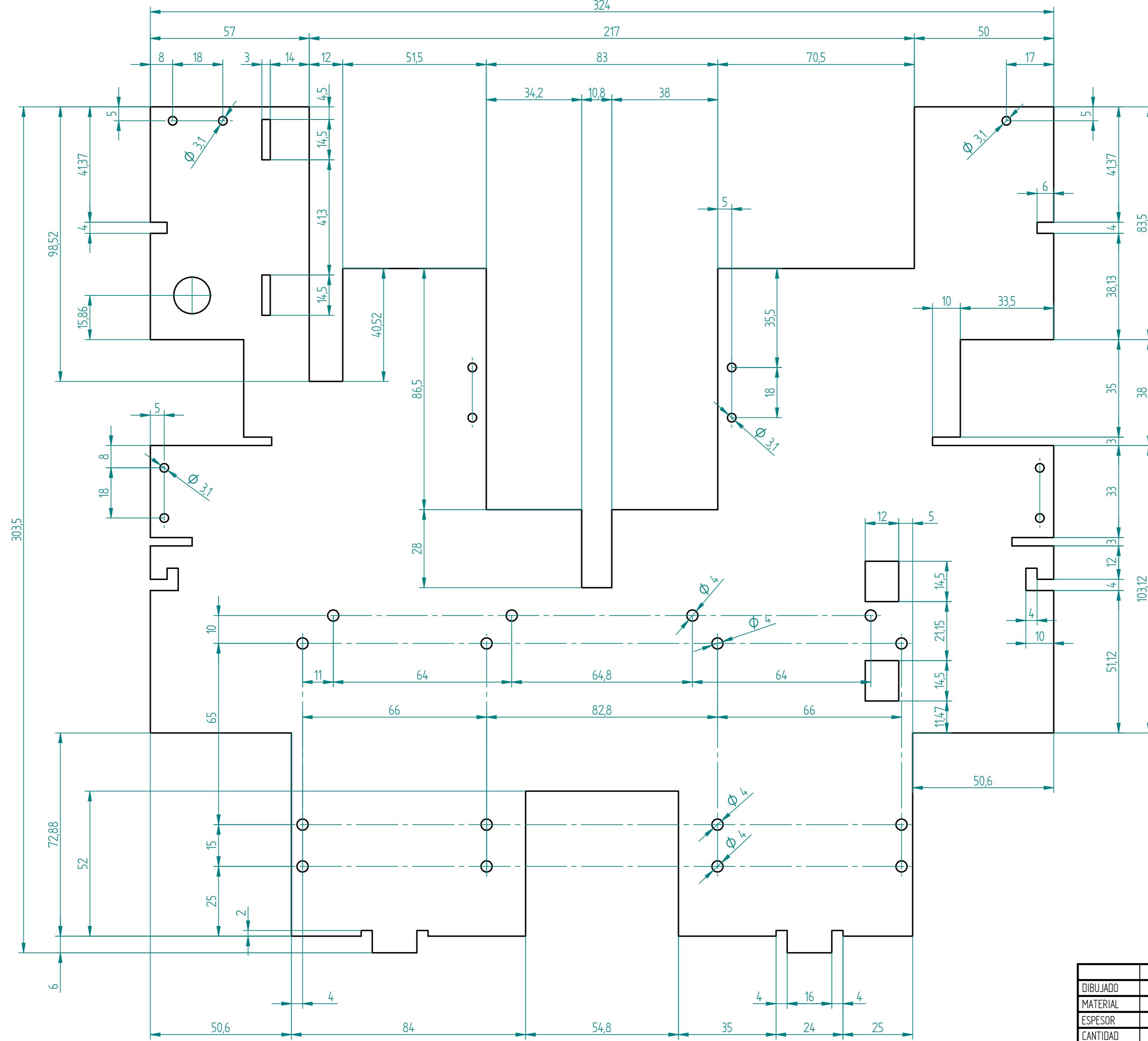
	NOMBRE	FECHA	Proyecto Eurobot 2015			
DIBUJADO	M.I.G	05-07-15	TITULO			
MATERIAL	TRESPA	TAPA SUPERIOR EXTRACTURA				
ESPESOR	6 MM					
CANTIDAD	1 UNIDADES					
DIMENSIONES ACOTADAS EN MILIMETROS						
TOLERANCIA GENERAL ±0.2						
ANGULO ±1°						
HOJA	NUMERO PLANO	00-01-02-014	REV	0		
A2						
ESCALA: 1:1	PESO:	HOJA 1 DE 1				





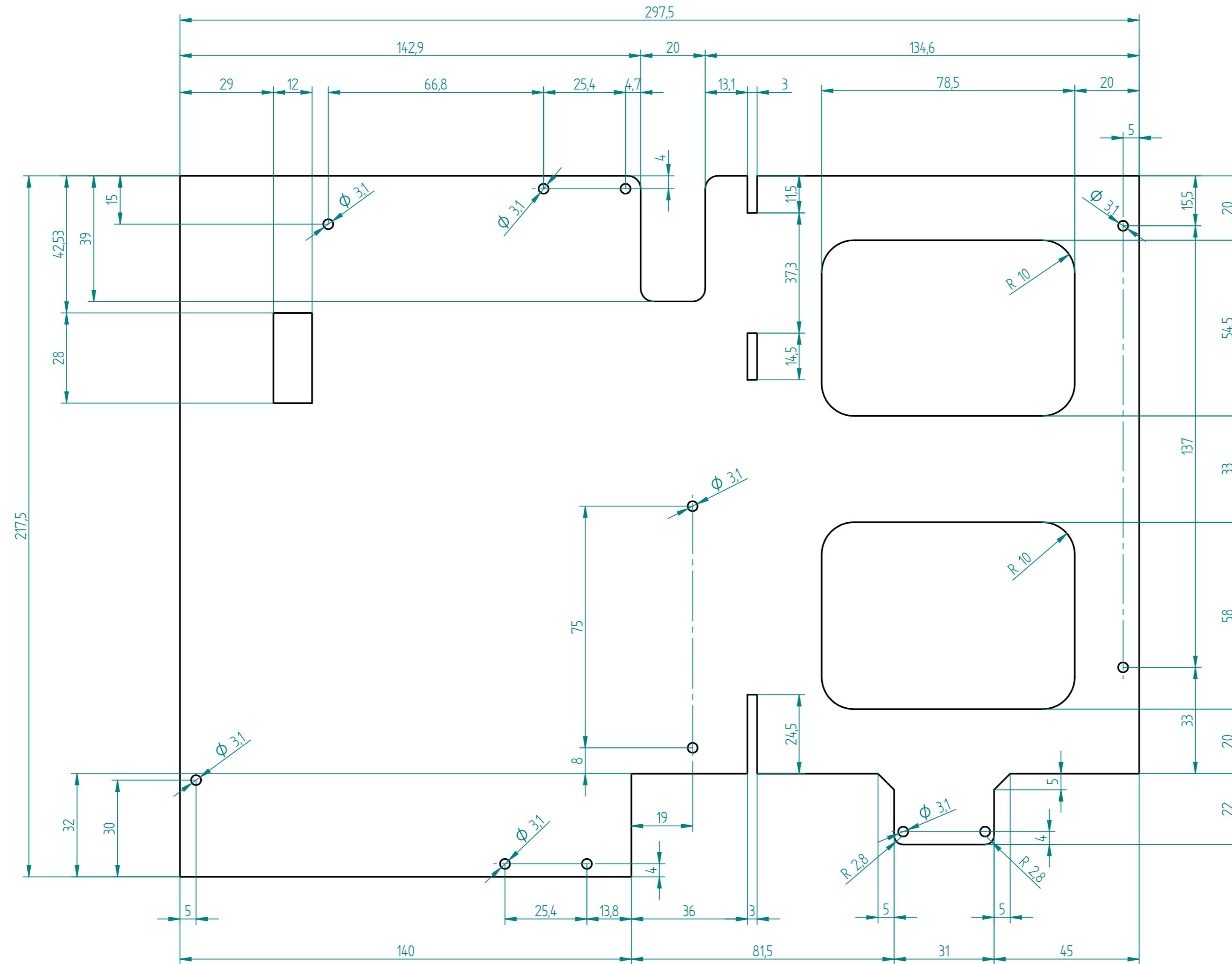
	NOMBRE	FECHA	Proyecto Eurobot 2015	
DIBUJADO	M.I.G	05-07-15		
MATERIAL	TRESPA	TITULO		
ESPEZOR	3 MM	ESTRUCTURA VERTICAL DELANTERO		
CANTIDAD	1 UNIDADES			
DIMENSIONES ACOTADAS EN MILIMETROS		HOJA	NUMERO PLANO	
TOLERANCIA GENERAL ±0.2		A2	00-01-03-007	REV 0
ANGULO ±1°		ESCALA:	PESO:	HOJA 1 DE 1





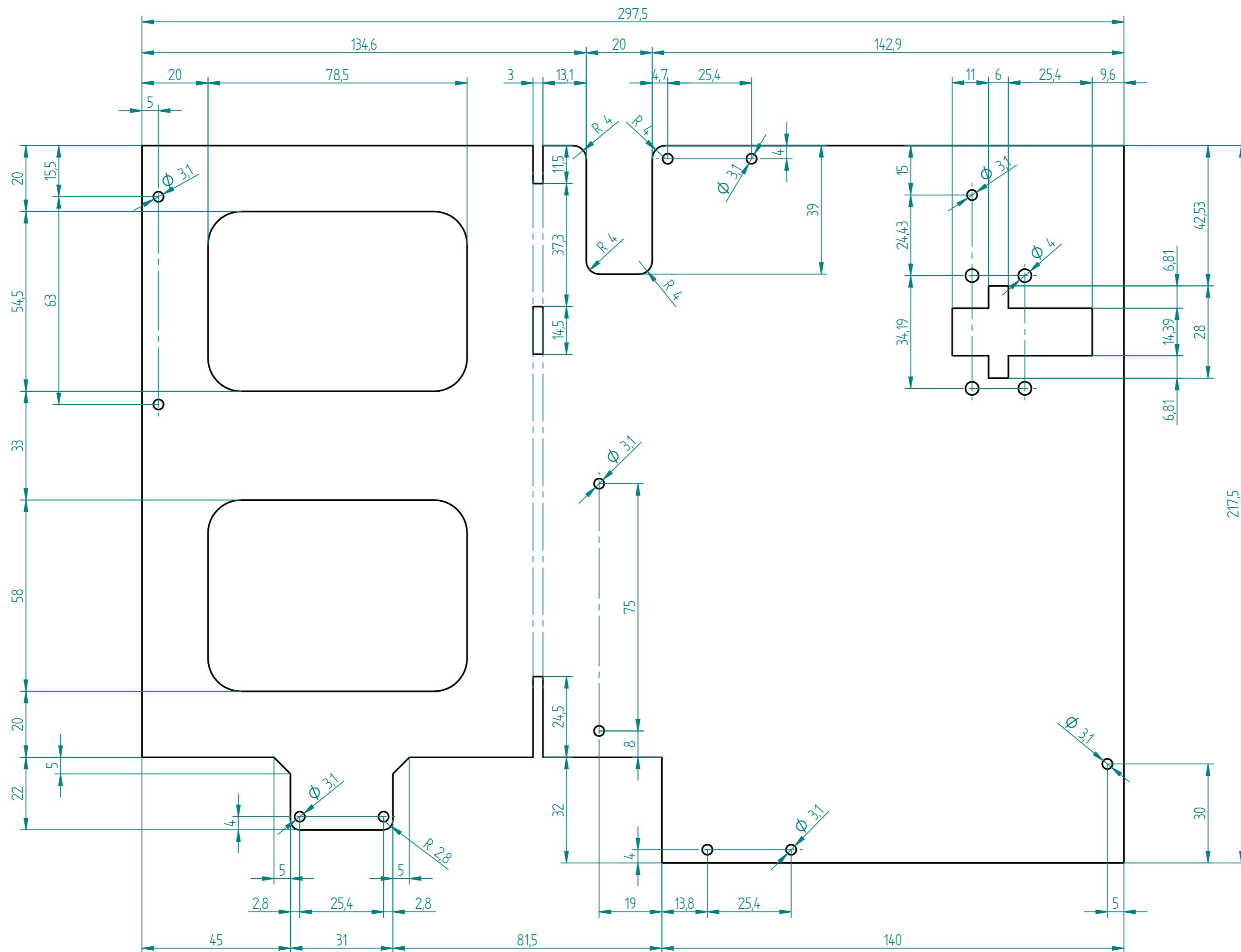
	NOMBRE	FECHA	Proyecto Eurobot 2015		
DIBUJADO	M.I.G	23-01-15			
MATERIAL	TRESPA	TITULO			
ESPESOR	3 MM				
CANTIDAD	1 UNIDADES				
DIMENSIONES ACOTADAS EN MILIMETROS TOLERANCIA GENERAL $\pm 0.2$ ANGULO $\pm 1^\circ$			HOJA	NUMERO PLANO	REV
			A2	00-01-04-008	0
			ESCALA:	PESO:	HOJA 1 DE 1





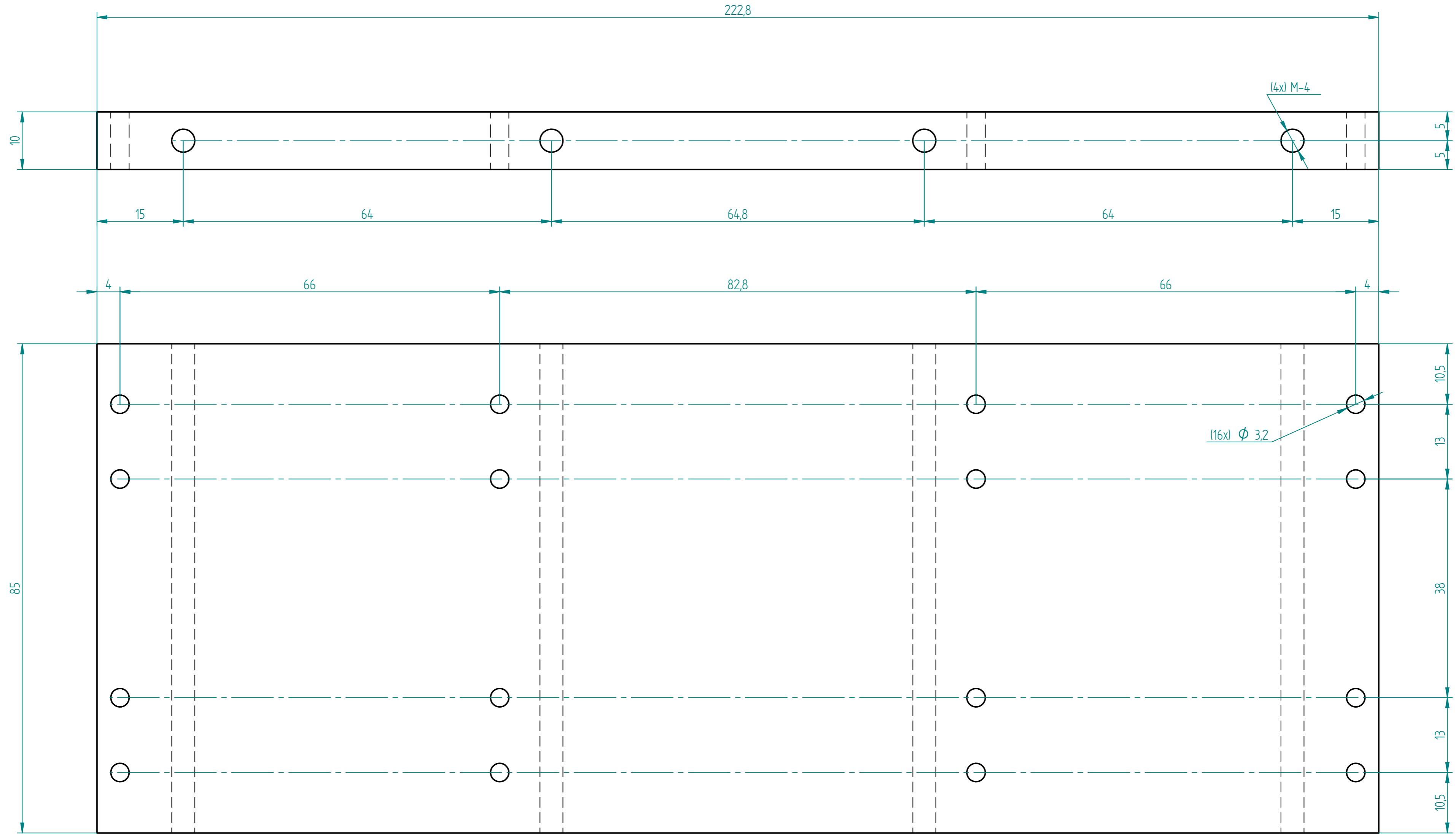
	NOMBRE	FECHA	Proyecto Eurobot 2015		
DIBUJADO	M.I.G	25-01-15			
MATERIAL	TRESPA				TITULO
ESPESOR	3 MM				ESTRUCTURA LATERAL IDQUIERDO
CANTIDAD	1 UNIDADES				
DIMENSIONES ACOTADAS EN MILIMETROS TOLERANCIA GENERAL ±0.2		HOJA	NUMERO PLANO 00-01-05-010		REV 0
ANGULO ±1°		ESCALA:	PESO:	HOJA 1 DE 1	





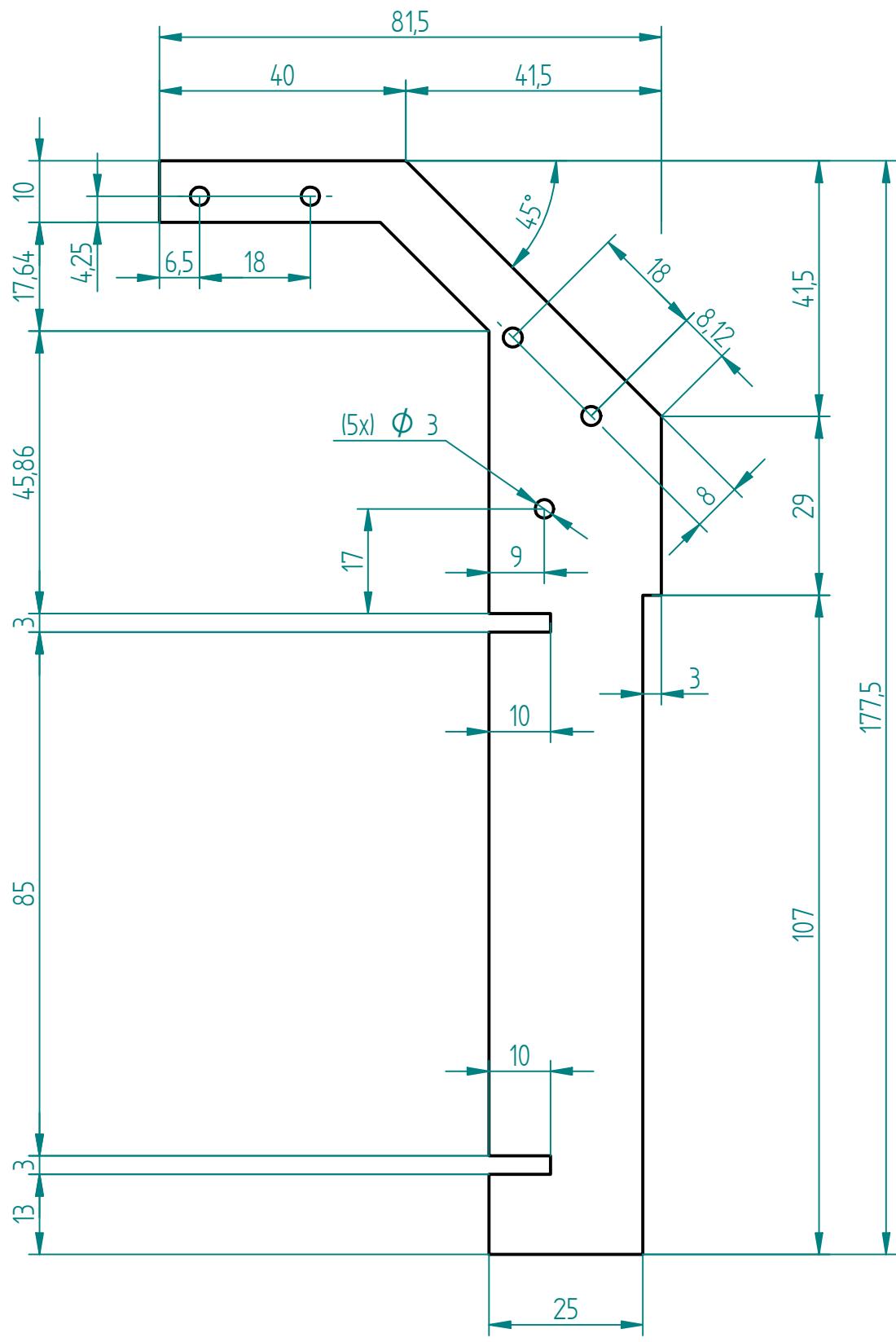
	NOMBRE	FECHA	Proyecto Eurobot 2015	
DIBUJADO	M.I.G	22-02-15		
MATERIAL	TRESPA			TITULO
ESPESOR	3 MM			ESTRUCTURA LATERAL DERECHO
CANTIDAD	1 UNIDADES			
DIMENSIONES ACOTADAS EN MILIMETROS		HOJA	NUMERO PLANO	
TOLERANCIA GENERAL ±0.2		A2	00-01-06-011	REV 0
ANGULO ±1°		ESCALA:	PESO:	HOJA 1 DE 1





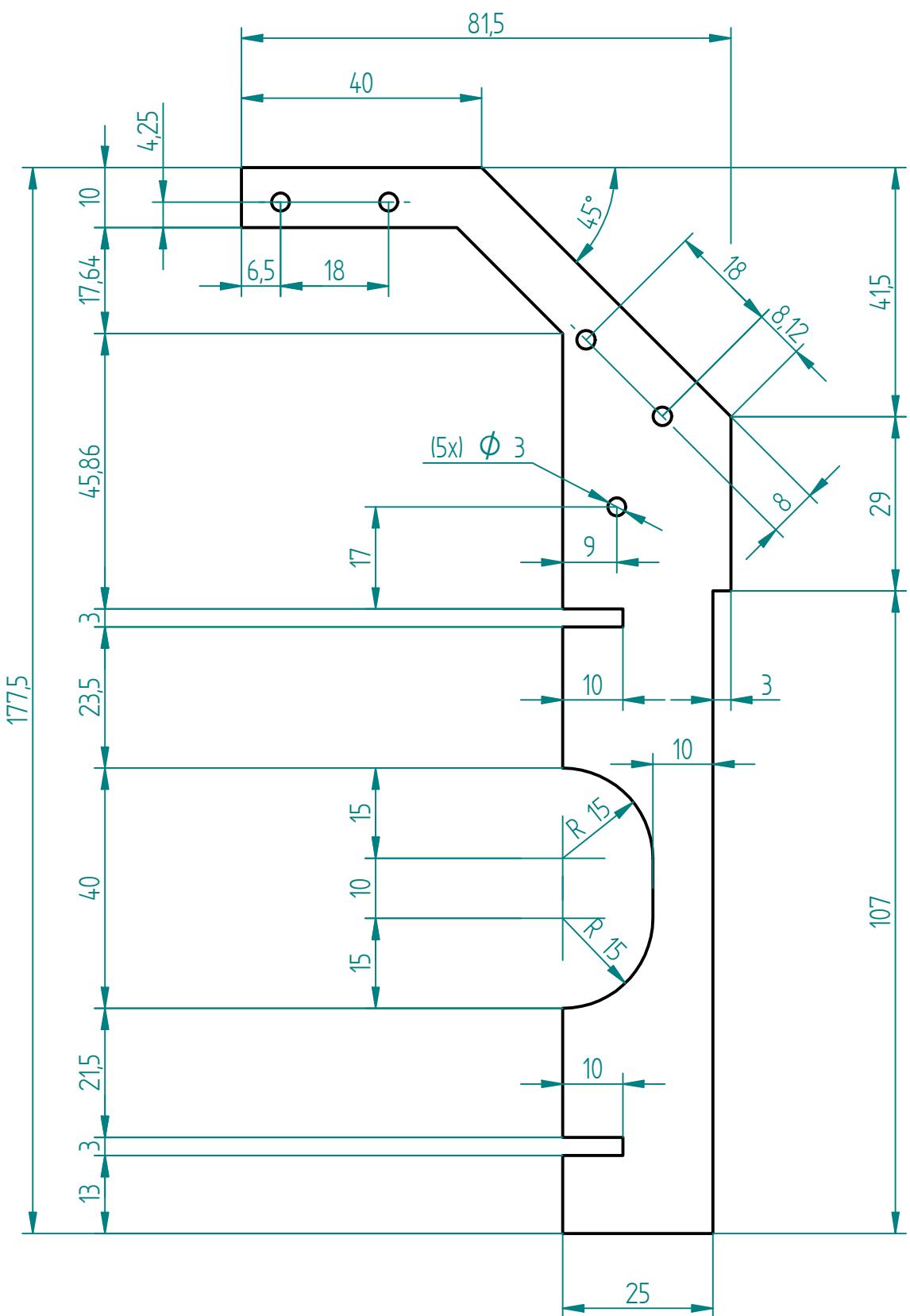
	NOMBRE	FECHA	Proyecto Eurobot 2015		
DIBUJADO	M.I.G	25-01-15			
MATERIAL	TRESPA		TITULO		
ESPESOR	10 MM		ESTRUCTURA TAPA BLOQUE MOTOR		
CANTIDAD	1 UNIDADES				
DIMENSIONES ACOTADAS EN MILIMETROS TOLERANCIA GENERAL ±0.2 ANGULO ±1°			HOJA A2	NUMERO PLANO 00-01-07-025	REV 0
			ESCALA: 1:1	PESO:	HOJA 1 DE 1





NOMBRE	M.I.G	FECHA	Proyecto Eurobot 2015		
DIBUJADO					
MATERIAL	TRESPA				
ESPESOR	3 MM				
CANTIDAD	1 UNIDADES		TITULO ESTRUCTURA SOPORTE SERVO		
DIMENSIONES ACOTADAS EN MILIMETROS					
TOLERANCIA GENERAL $\pm 0.2$			HOJA	NUMERO PLANO	REV
			A4	00-01-08-036	0
ANGULO $\pm 1^\circ$			ESCALA: 1:1	PESO:	HOJA 1 DE 1



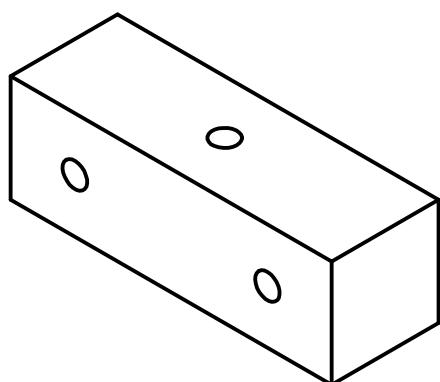
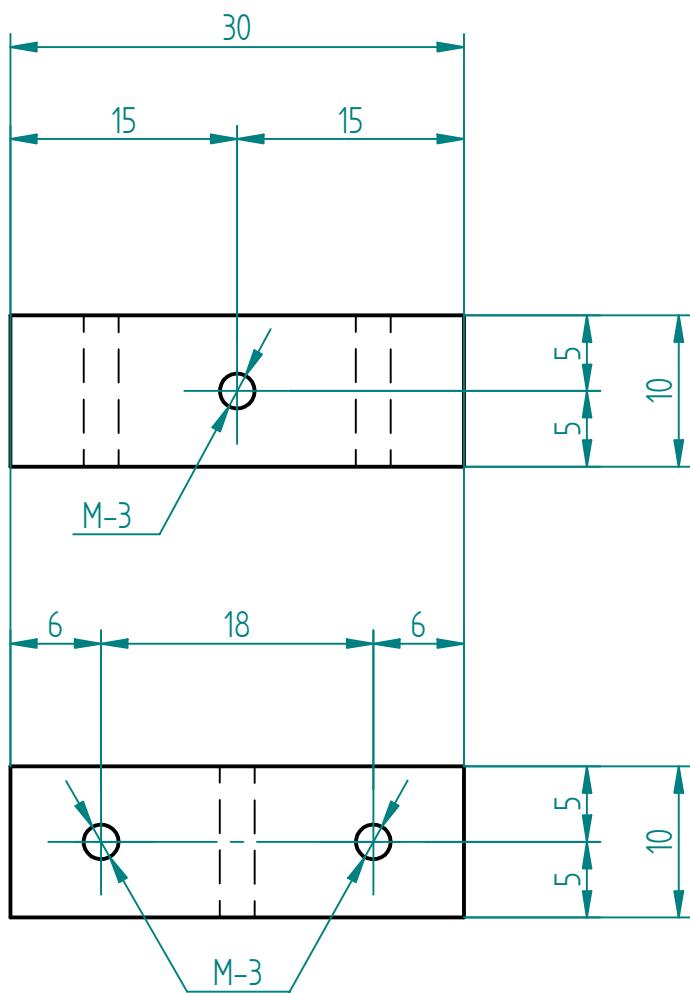


DIBUJADO	NOMBRE	FECHA
	M.I.G	21-02-15
MATERIAL	TRESPA	
ESPESOR	3 MM	
CANTIDAD	1 UNIDADES	
DIMENSIONES ACOTADAS EN MILIMETROS		
TOLERANCIA GENERAL $\pm 0.2$		
ANGULO $\pm 1^\circ$		

Proyecto Eurobot 2015

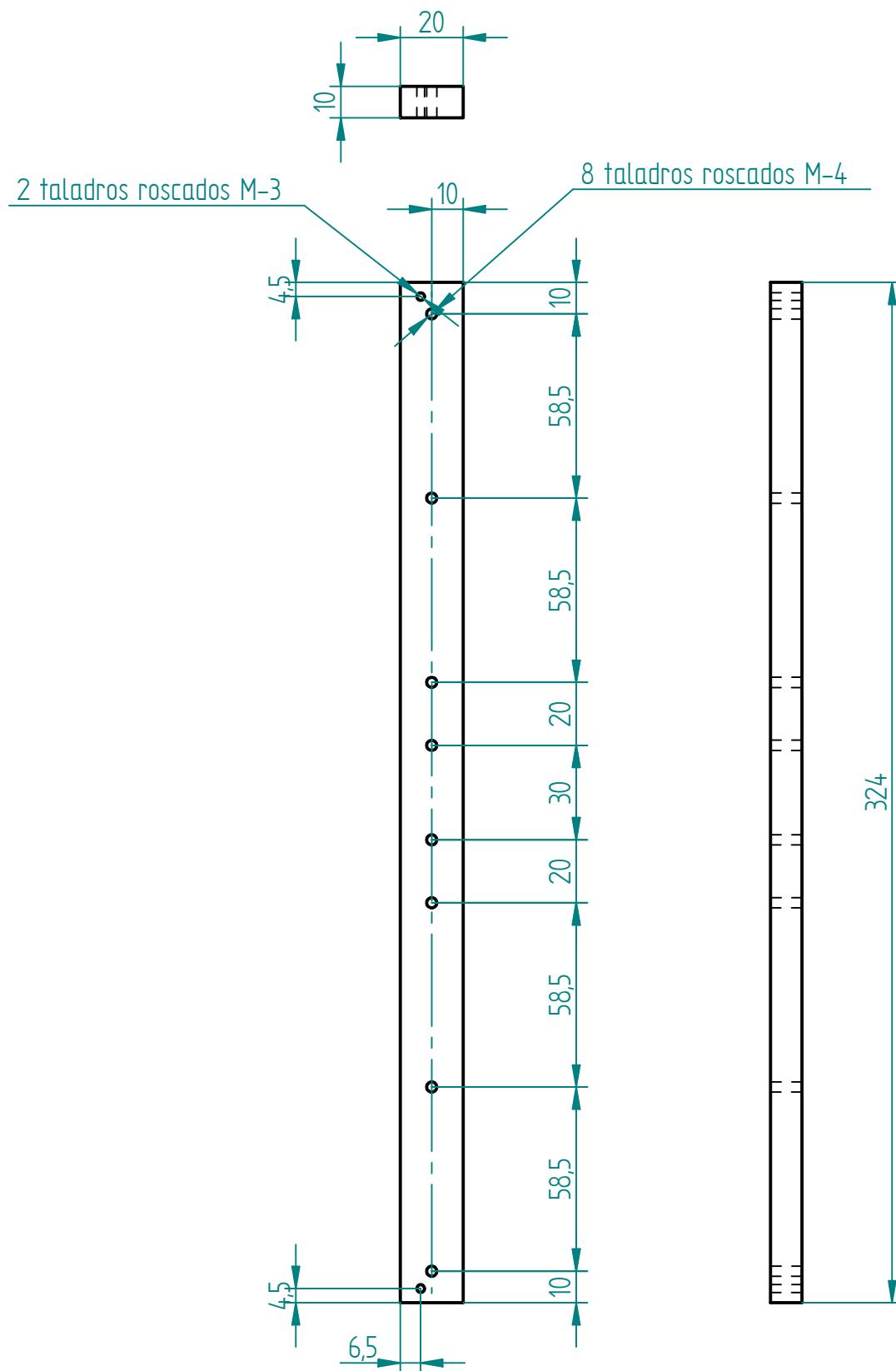
TITULO	SOporte SERVO BRAZO PALOMITAS DERECHO	REV 0
HOJA	NUMERO PLANO	
A4	00-01-09-038	
ESCALA:	1:1	PESO:
		HOJA 1 DE 1





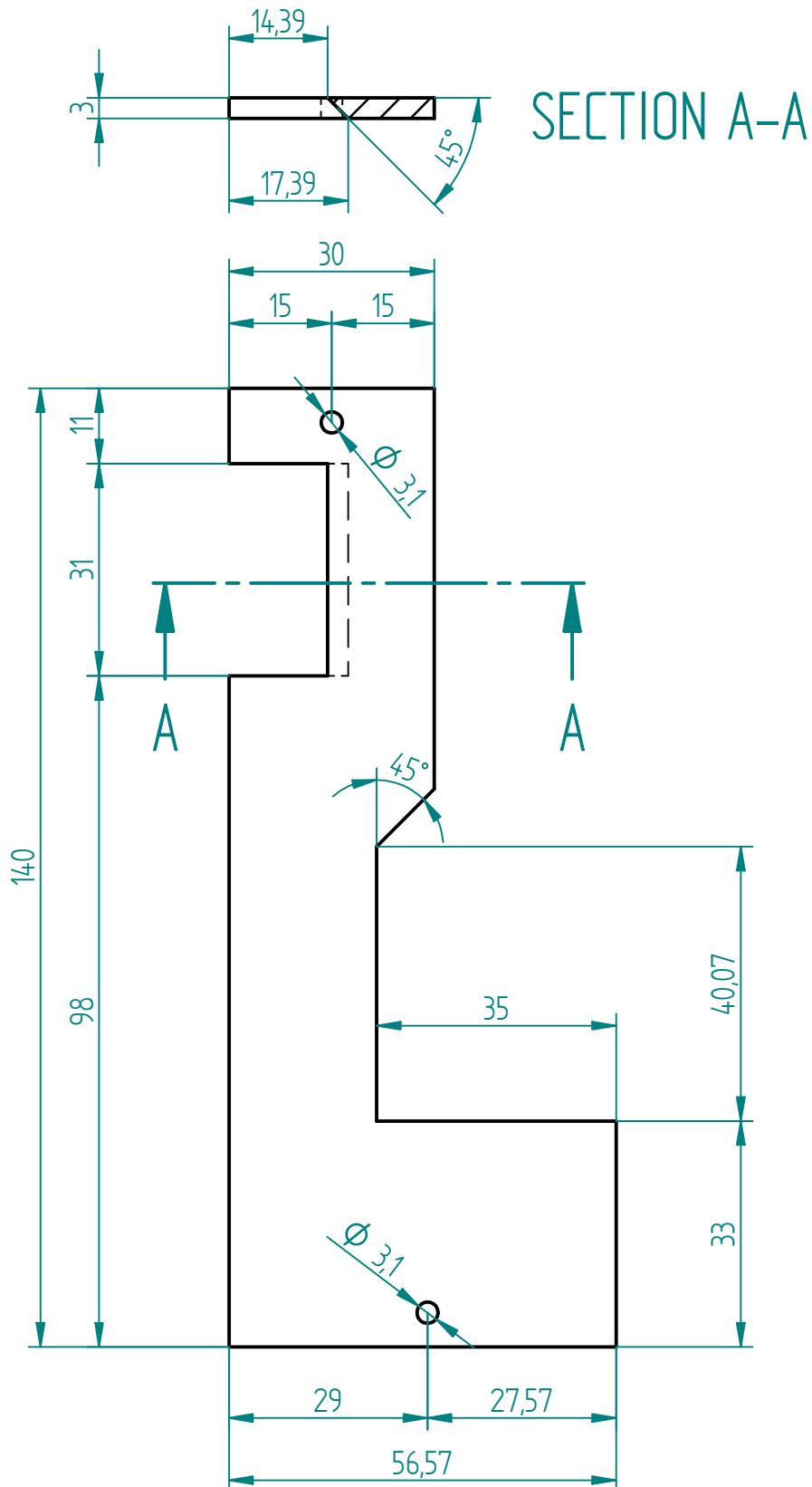
NOMBRE		FECHA	Proyecto Eurobot 2015		
DIBUJADO	M.I.G	22-02-15			
MATERIAL	ALUMINIO			TITULO	
ESPESOR	10 x 10 MM			TACOS ALUMINIO	
CANTIDAD	23 UNIDADES				
DIMENSIONES ACOTADAS EN MILIMETROS TOLERANCIA GENERAL $\pm 0.2$ ANGULO $\pm 1^\circ$			HOJA	NUMERO PLANO	REV
			A4	00-01-10-084	0
			ESCALA: 2:1		PESO:
			HOJA 1 DE 1		





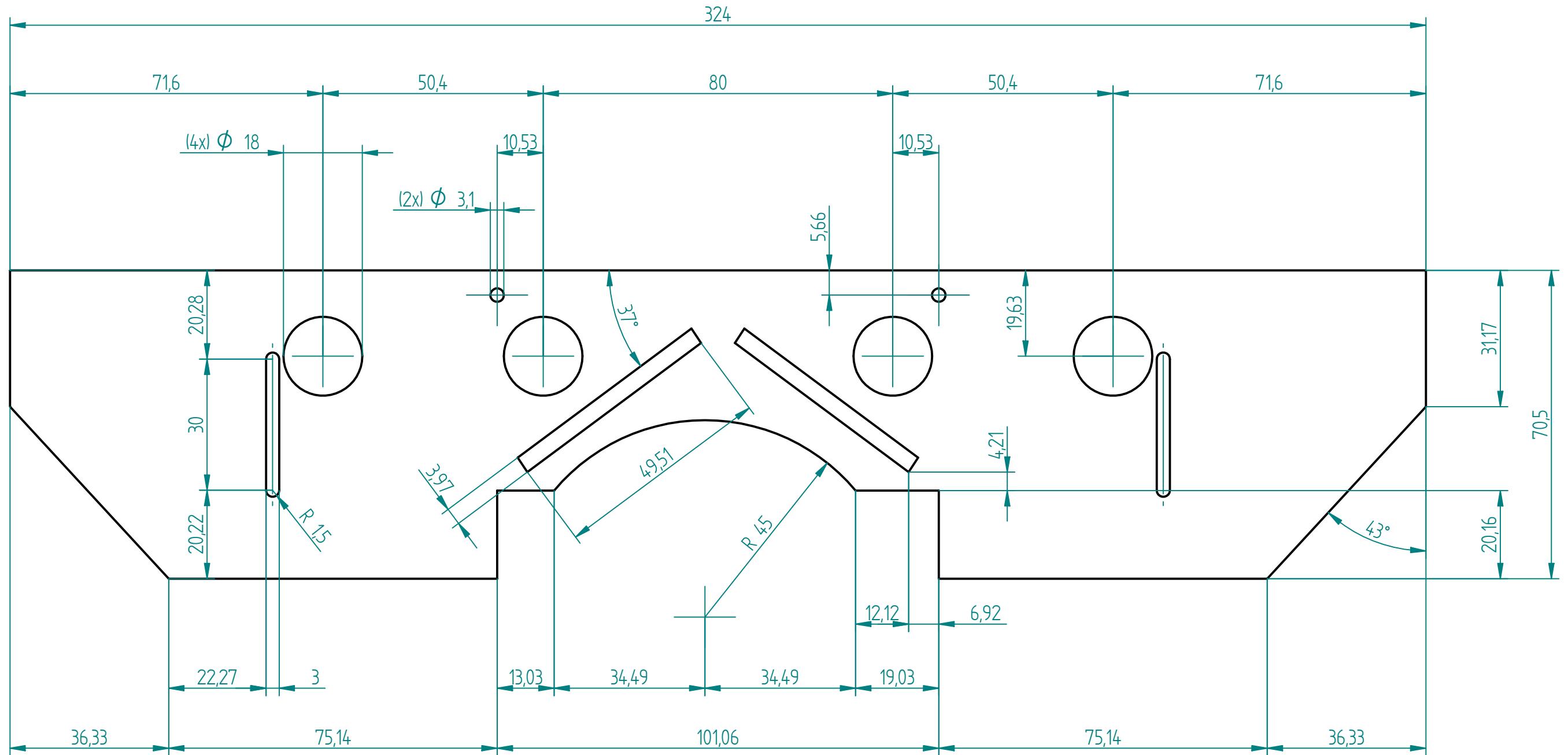
NOMBRE	M.I.G	FECHA	Proyecto Eurobot 2015		
DIBUJADO					
MATERIAL	ACERO F-100		TITULO		
ESPESOR			REFUERZO DELANTERO BASE		
CANTIDAD	1 UNIDADES				
DIMENSIONES ACOTADAS EN MILIMETROS			HOJA	NUMERO PLANO	REV
TOLERANCIA GENERAL $\pm 0.2$			A4	00-01-12-003	0
ANGULO $\pm 1^\circ$			ESCALA: 1:2	PESO:	HOJA 1 DE 1





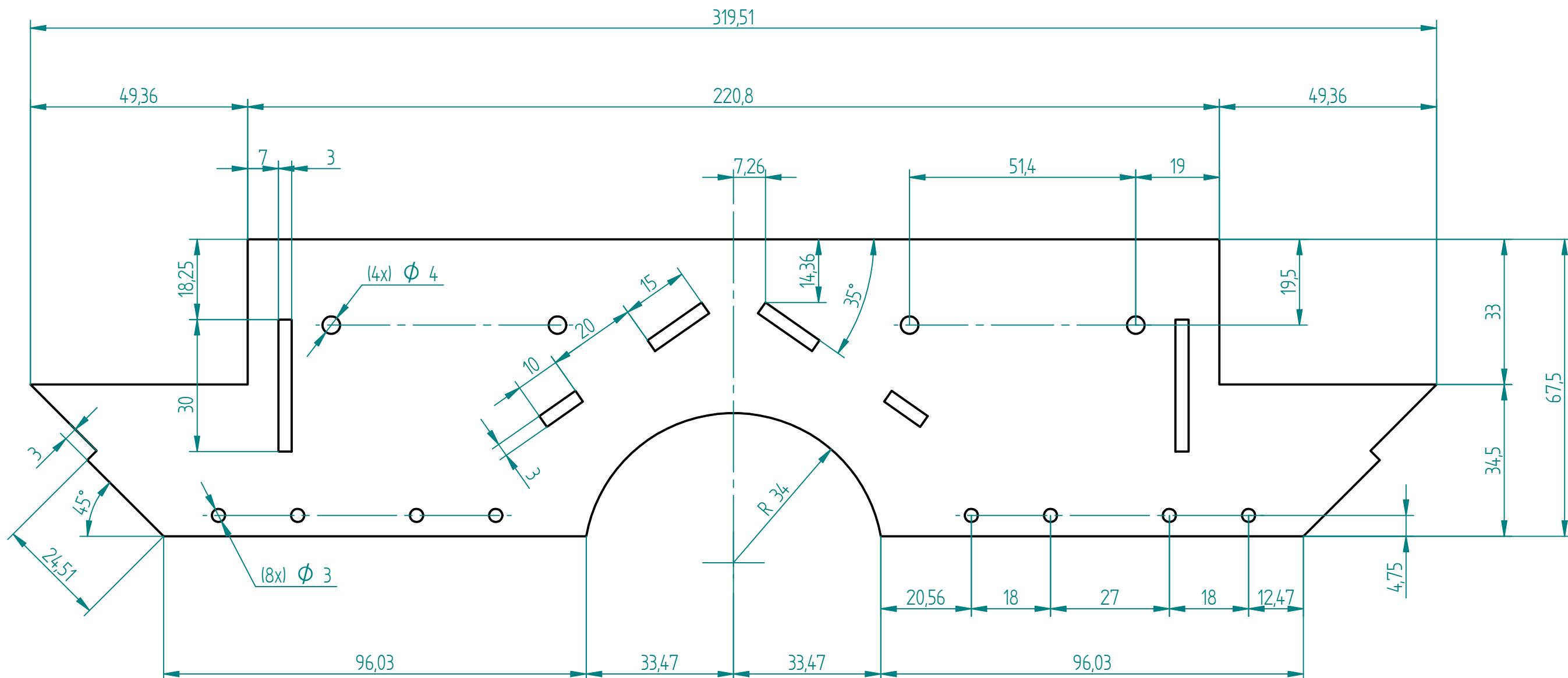
NOMBRE	M.I.G	FECHA	Proyecto Eurobot 2015		
DIBUJADO					
MATERIAL	TRESPA		TITULO		
ESPESOR	3 MM		ESTRUCTURA CHAFLAN TRASERO		
CANTIDAD	1 S/D + 1 M/C				
DIMENSIONES ACOTADAS EN MILIMETROS TOLERANCIA GENERAL ±0.2 ANGULO ±1°			HOJA	NUMERO PLANO	REV
			A4	00-01-12-012	0
			ESCALA: 1:1	PESO:	HOJA 1 DE 1





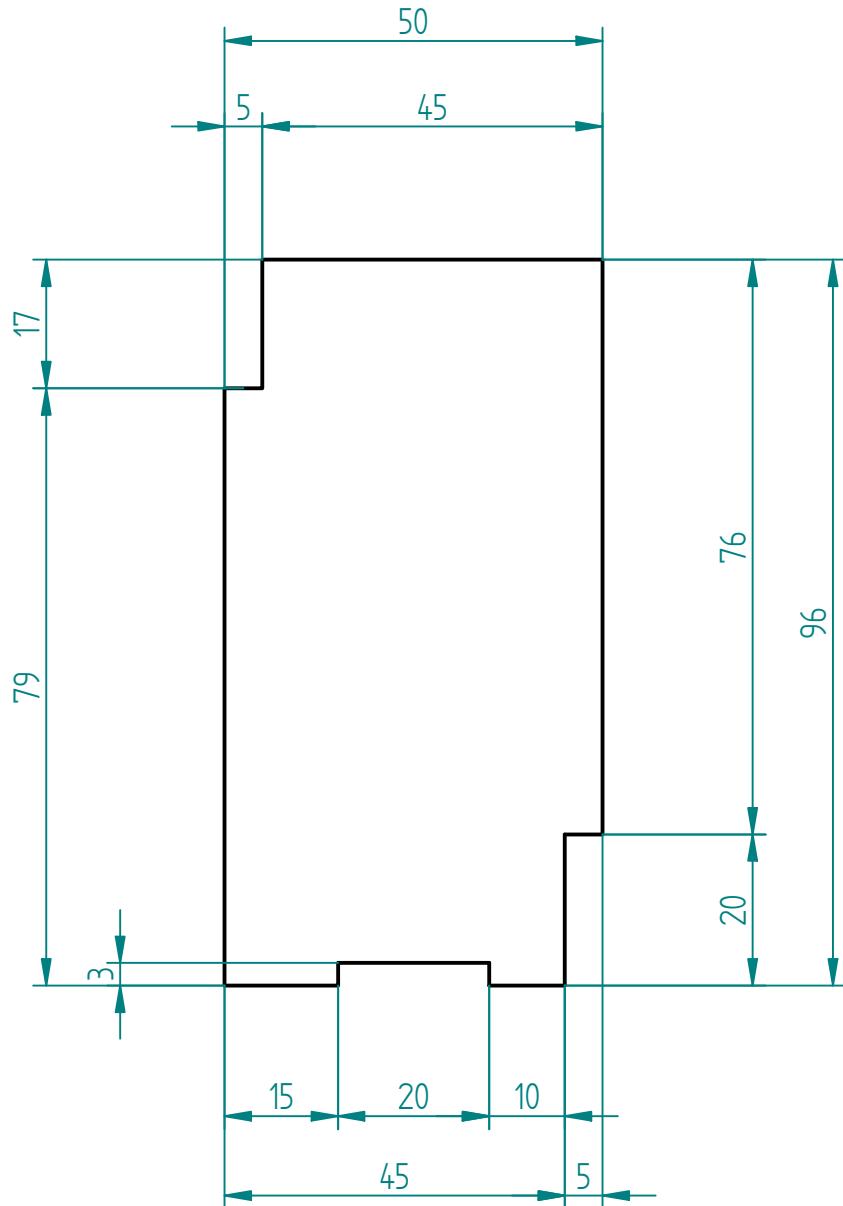
	NOMBRE	FECHA	Proyecto Eurobot 2015	
DIBUJADO	M.I.G	24-01-15		
MATERIAL	TRESPA			
ESPESOR	3 MM			
CANTIDAD	1 UNIDADES			
DIMENSIONES ACOTADAS EN MILIMETROS			HOJA A3	NUMERO PLANO 00-01-13-027
TOLERANCIA GENERAL ±0.2			REV 0	
ANGULO ±1°			ESCALA: 1:1	PESO: HOJA 1 DE 1





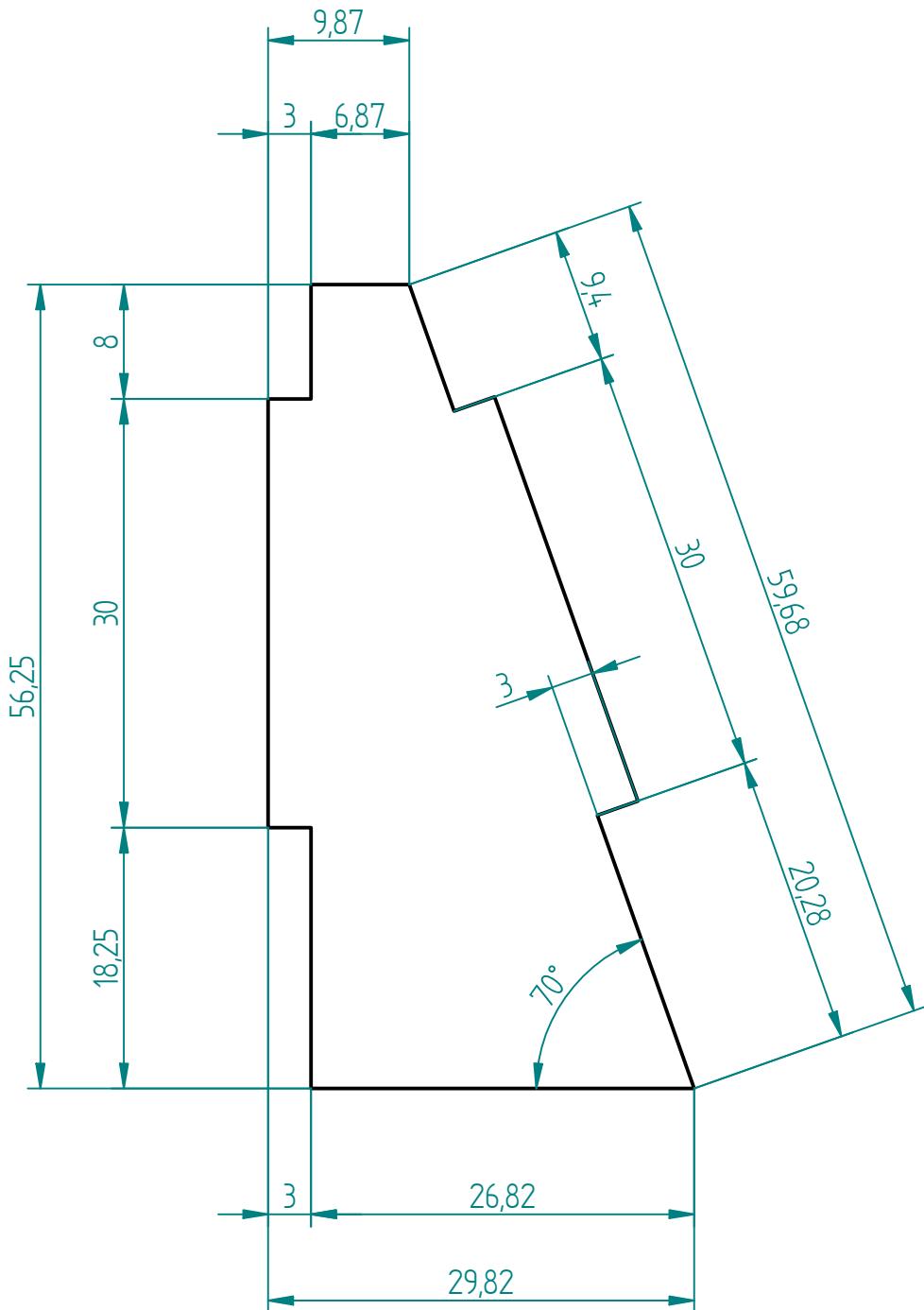
	NOMBRE	FECHA	Proyecto Eurobot 2015		
DIBUJADO	M.I.G	23-01-15			
MATERIAL	TRESPA				
ESPESOR	3 MM				
CANTIDAD	1 UNIDADES				
DIMENSIONES ACOTADAS EN MILIMETROS TOLERANCIA GENERAL ±0.2  ANGULO ±1°			HOJA	NUMERO PLANO	REV
			A3	00-01-14-026	0
			ESCALA:	1:1	PESO:
			HOJA 1 DE 1		





NOMBRE	M.I.G	FECHA	Proyecto Eurobot 2015		
DIBUJADO		24-01-15			
MATERIAL	TRESPA		TITULO		
ESPESOR	3 MM		ESTRUCTURA PARED VASO		
CANTIDAD	2 UNIDADES				
DIMENSIONES ACOTADAS EN MILIMETROS TOLERANCIA GENERAL ±0.2 ANGULO ±1°			HOJA	NUMERO PLANO	REV
			A4	00-01-15-028	0
			ESCALA: 1:1	PESO:	HOJA 1 DE 1



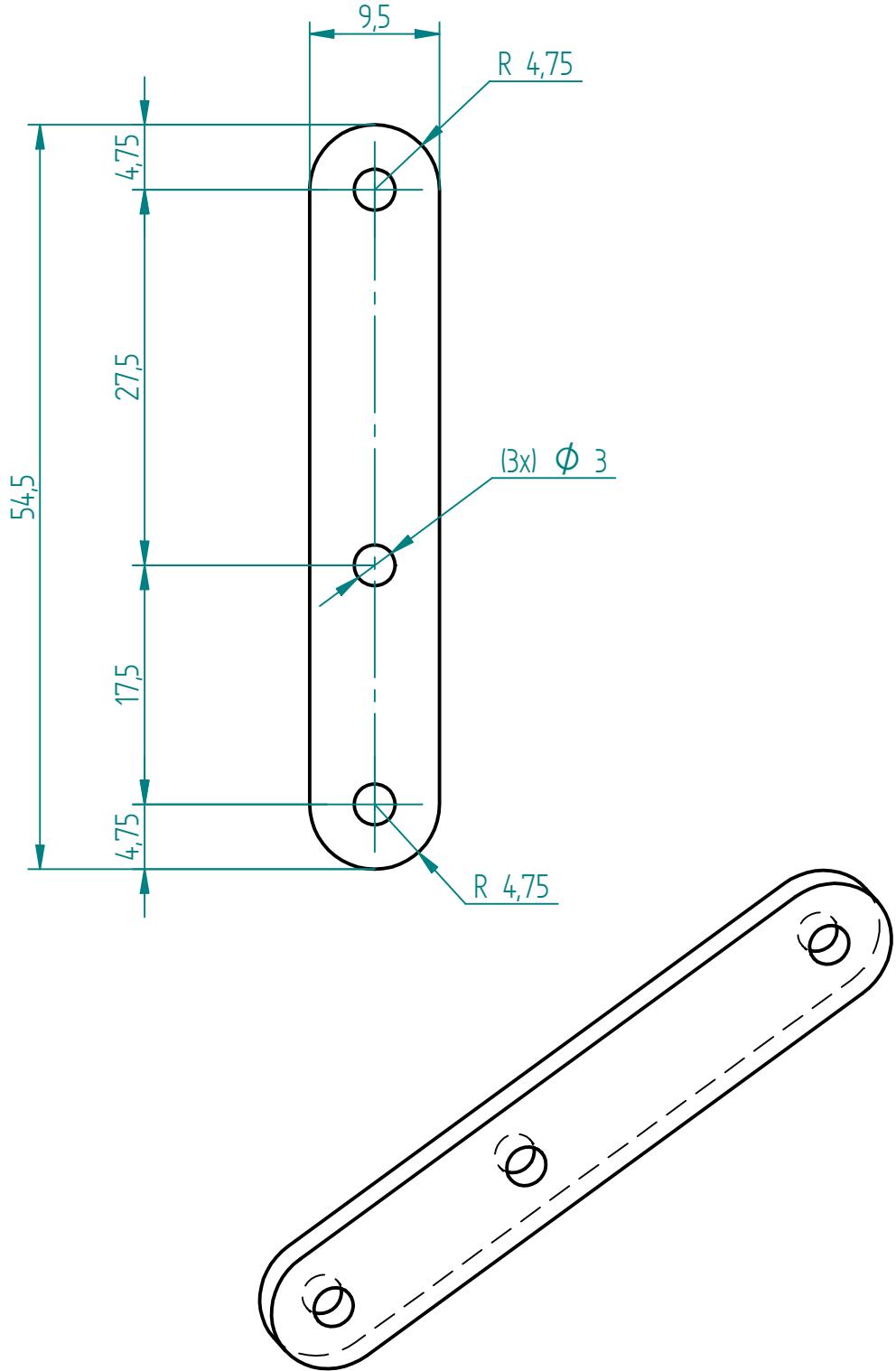


	NOMBRE	FECHA
DIBUJADO	M.I.G	24-01-15
MATERIAL	TRESPA	
ESPESOR	3 MM	
CANTIDAD	2 UNIDADES	
DIMENSIONES ACOTADAS EN MILIMETROS		
TOLERANCIA GENERAL $\pm 0.2$		
ANGULO $\pm 1^\circ$		

## Proyecto Eurobot 2015

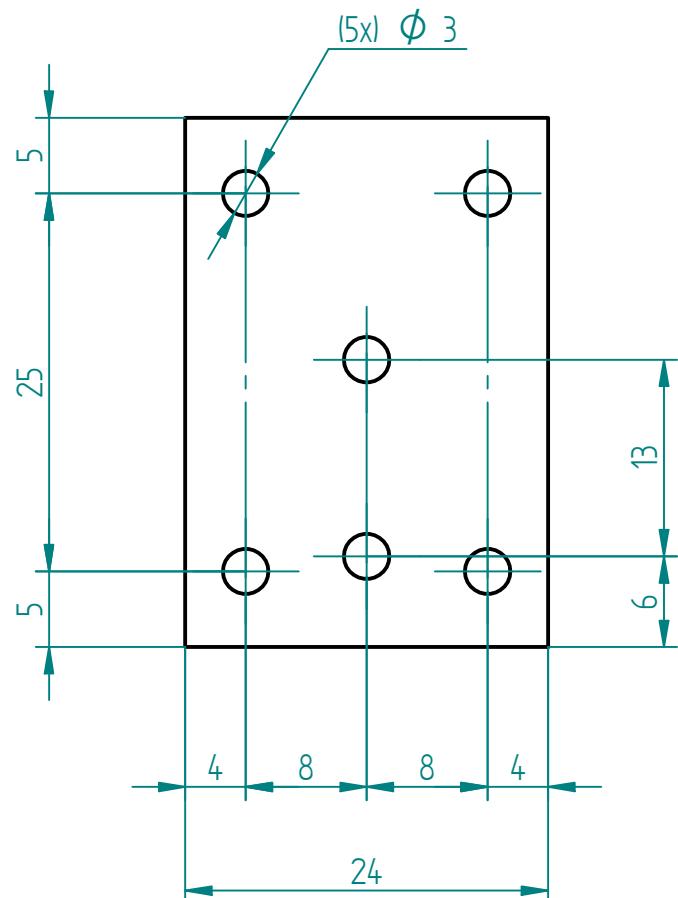
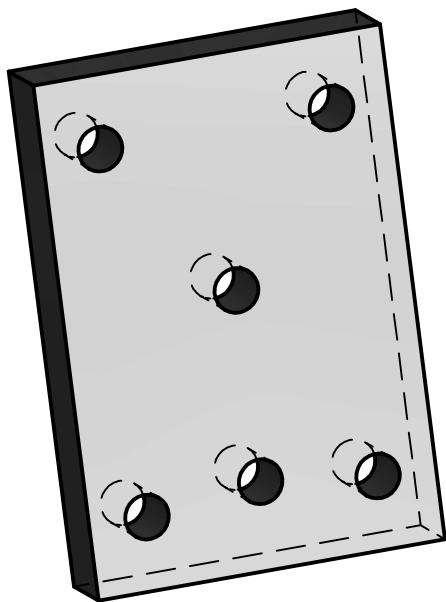
TITULO		
ESTRUCTURA PENDIENTE PISO TRASERO		
HOJA	NUMERO PLANO	REV
A4	00-01-16-029	0
ESCALA: 2:1	PESO:	HOJA 1 DE 1





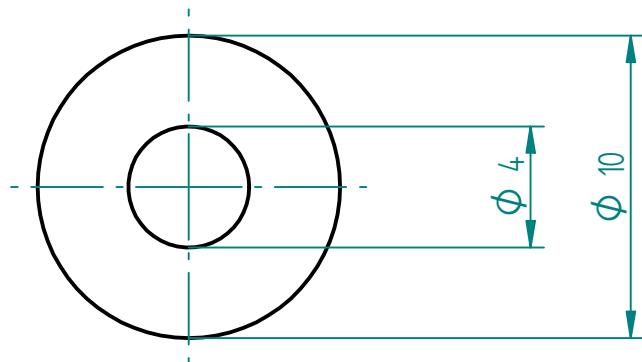
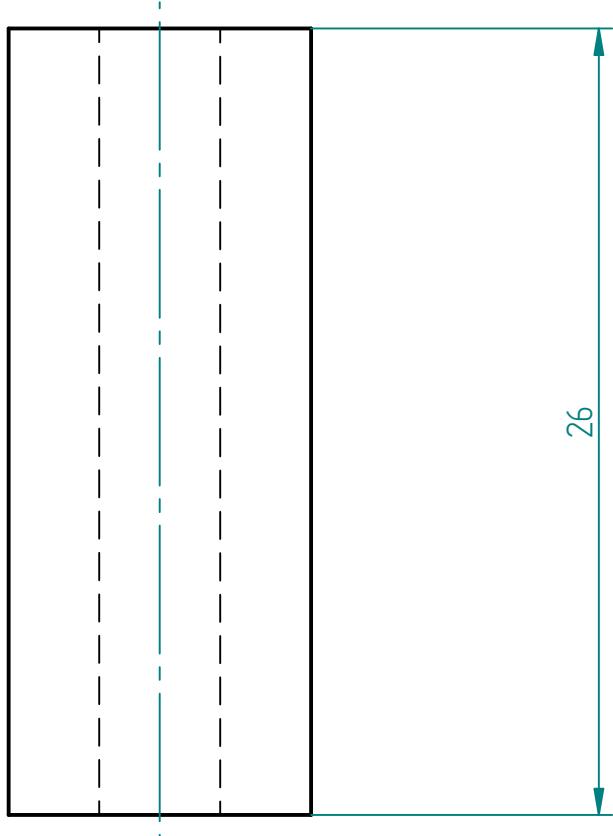
NOMBRE		FECHA	Proyecto Eurobot 2015		
DIBUJADO	M.I.G	24-02-15			
MATERIAL	TRESPA				
ESPESOR	3 MM				
CANTIDAD	2 UNIDADES				
DIMENSIONES ACOTADAS EN MILIMETROS TOLERANCIA GENERAL ±0.2 ANGULO ±1°			HOJA A4	NUMERO PLANO 00-01-17-089	REV 0
			ESCALA: 2:1	PESO:	HOJA 1 DE 1





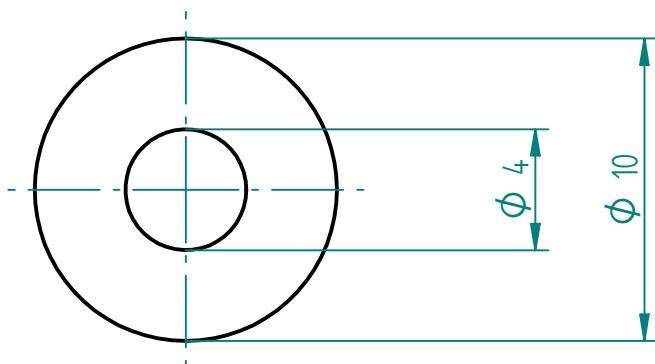
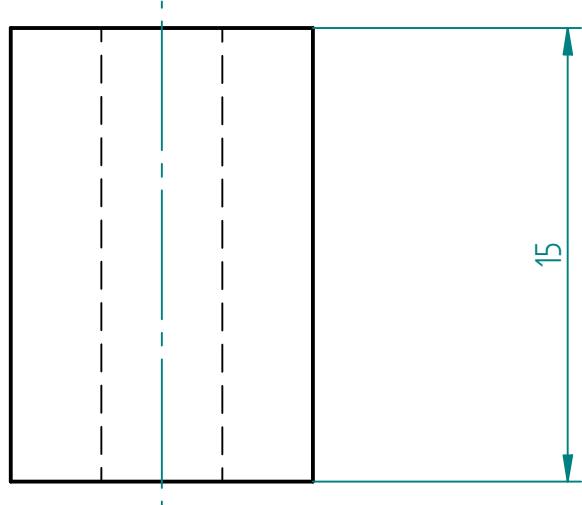
NOMBRE	FECHA		Proyecto Eurobot 2015		
DIBUJADO	M.I.G	01-03-15			
MATERIAL	TRESPA			TITULO	
ESPESOR	6 MM			SOPORTE TEFLON APOYO TRASERO	
CANTIDAD	2 UNIDADES				
DIMENSIONES ACOTADAS EN MILIMETROS TOLERANCIA GENERAL $\pm 0.2$ ANGULO $\pm 1^\circ$			HOJA	NUMERO PLANO	REV
			A4	00-01-18-203	0
			ESCALA:	PESO:	HOJA 1 DE 1





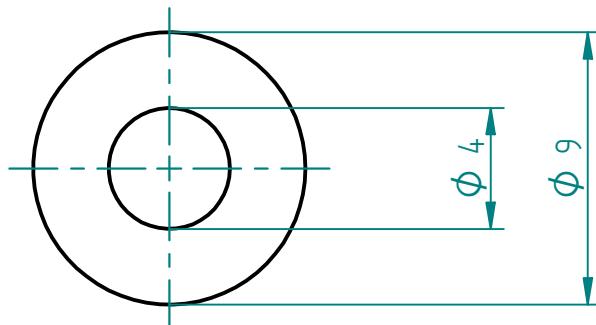
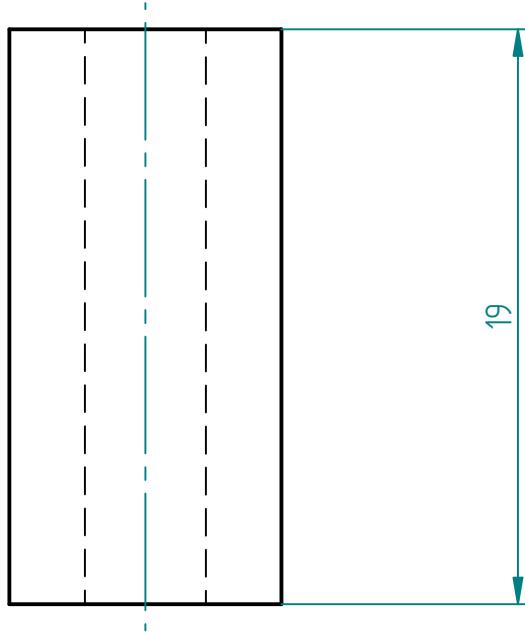
NOMBRE	FECHA	Proyecto Eurobot 2015		
DIBUJADO	M.I.G	22-02-15		
MATERIAL	ALUMINIO DURAL			TITULO
ESPESOR				SEPARADOR TRASERO BASE CARROS
CANTIDAD	4 UNIDADES			
DIMENSIONES ACOTADAS EN MILIMETROS		HOJA	NUMERO PLANO	REV
TOLERANCIA GENERAL $\pm 0.2$		A4	00-01-19-054	0
ANGULO $\pm 1^\circ$		ESCALA: 4:1	PESO:	HOJA 1 DE 1





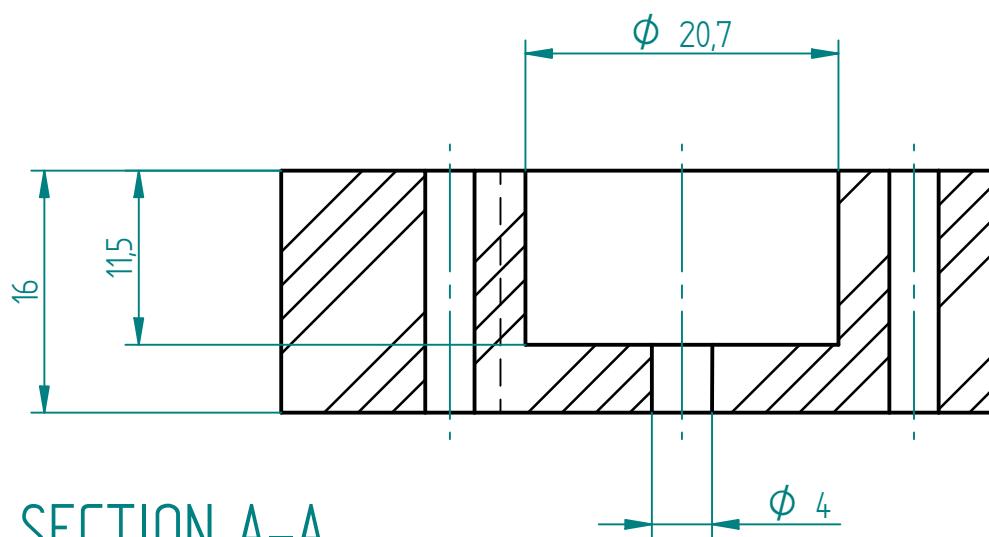
NOMBRE	FECHA	Proyecto Eurobot 2015		
DIBUJADO	M.I.G	21-02-15		
MATERIAL	ALUMINIO DURAL			TITULO
ESPESOR				SEPARADOR LASTRE Y BASE CARROS DEL.
CANTIDAD	4 UNIDADES			
DIMENSIONES ACOTADAS EN MILIMETROS		HOJA	NUMERO PLANO	REV
TOLERANCIA GENERAL ±0.2		A4	00-01-20-055	0
ANGULO ±1°		ESCALA:	4:1	PESO:
				HOJA 1 DE 1



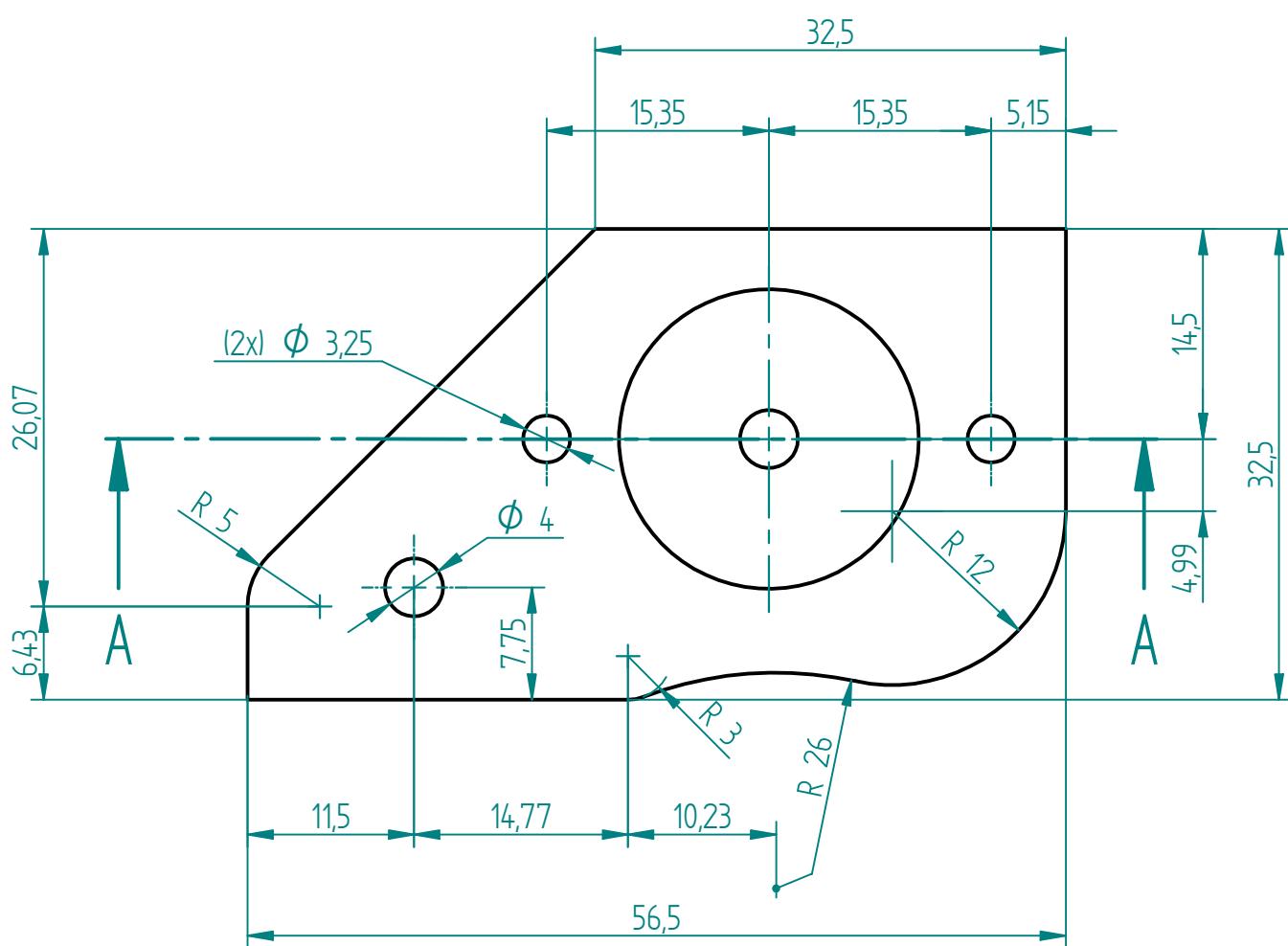


NOMBRE	FECHA	Proyecto Eurobot 2015		
DIBUJADO	17-02-15			
MATERIAL	ALUMINIO DURAL			
ESPESOR				
CANTIDAD	2 UNIDADES	TITULO SEPARADOR BASE Y CARROS DELANTEROS		
DIMENSIONES ACOTADAS EN MILIMETROS TOLERANCIA GENERAL $\pm 0.2$ ANGULO $\pm 1^\circ$		HOJA	NUMERO PLANO 00-01-21-053	REV 0
		A4		
		ESCALA: 4:1	PESO:	HOJA 1 DE 1





SECTION A-A



	NOMBRE	FECHA
DIBUJADO	M.I.G	21-02-15
MATERIAL	TRESPA	
ESPESOR	16 MM	
CANTIDAD	1 S/D + A M/C	
DIMENSIONES ACOTADAS EN MILIMETROS		
TOLERANCIA GENERAL $\pm 0.2$		
ANGULO $\pm 1^\circ$		

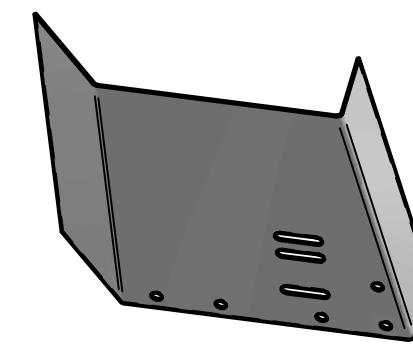
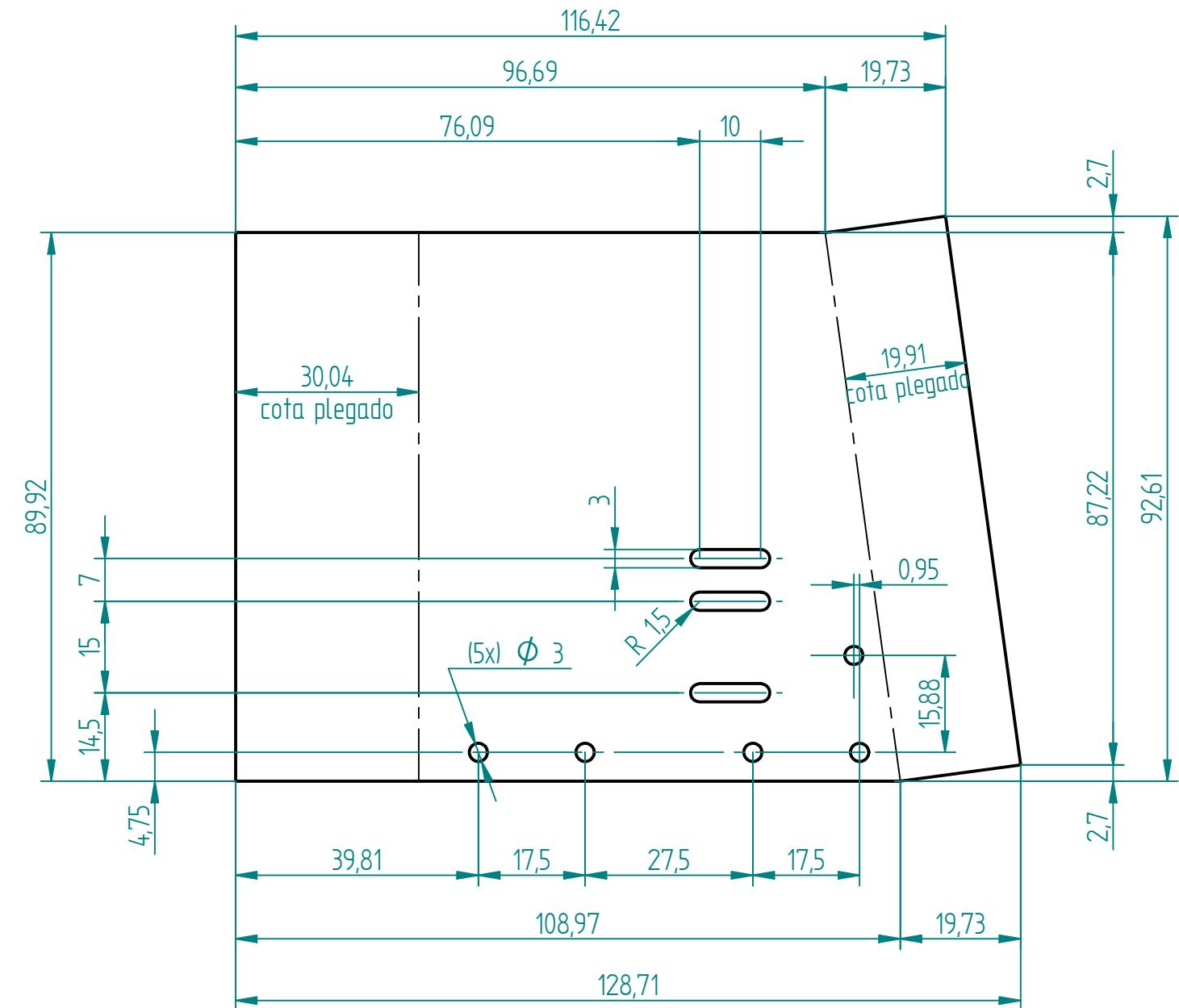
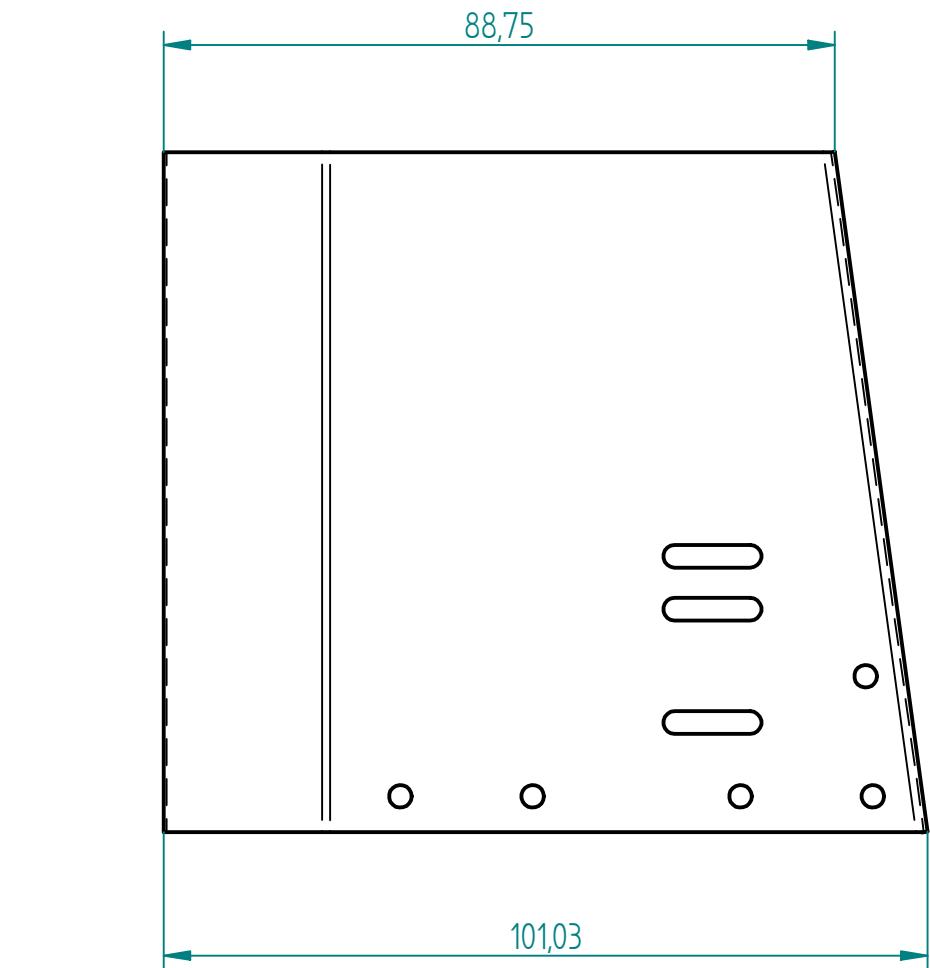
Proyecto Eurobot 2015

TITULO

SOPORTE ROLLON

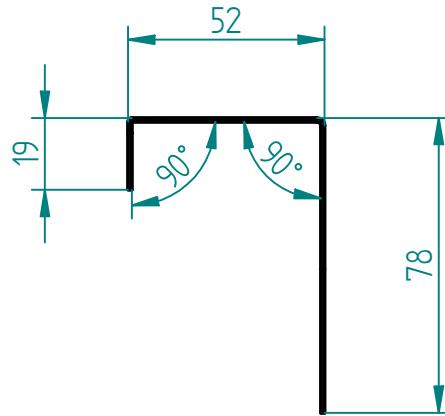
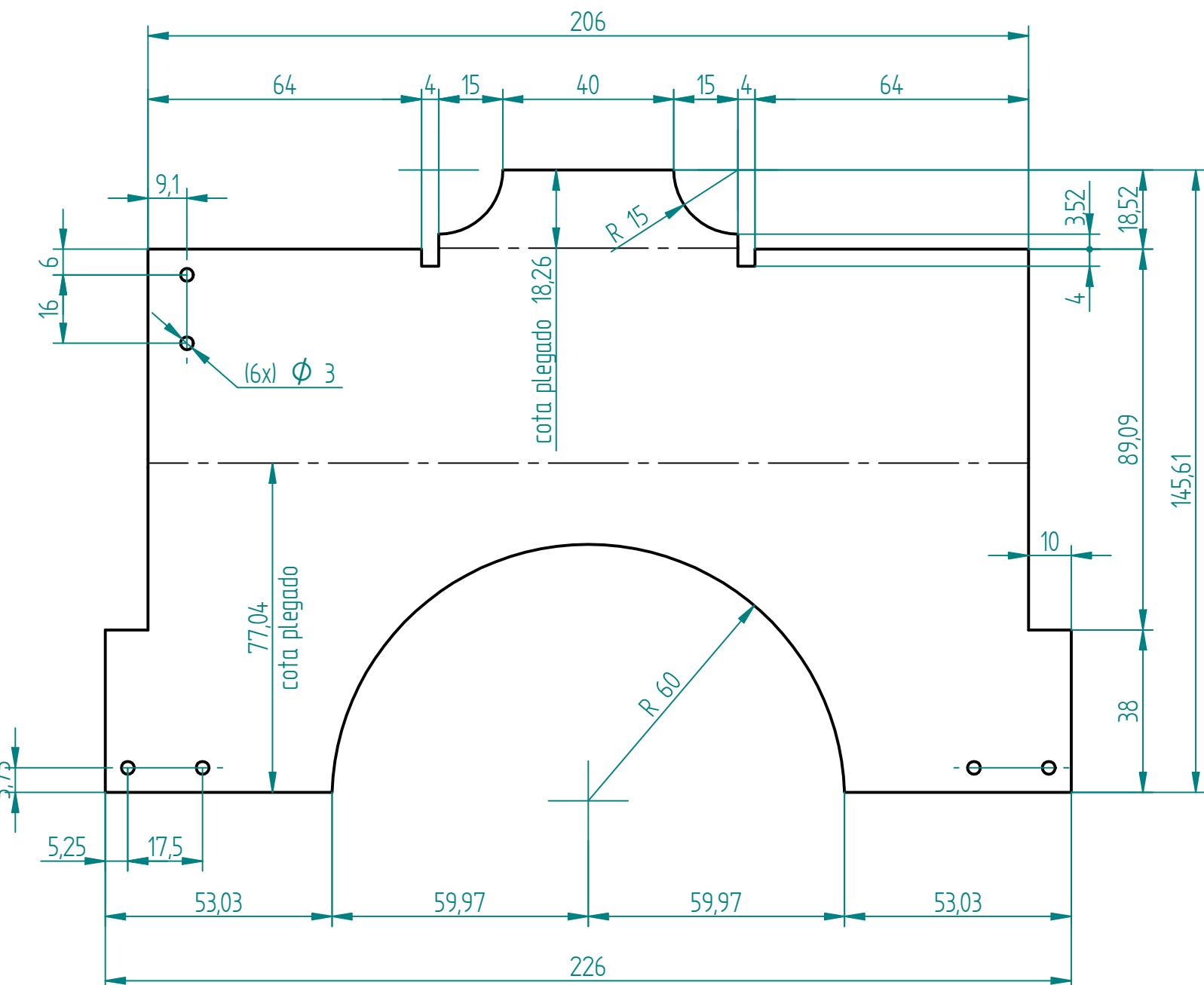
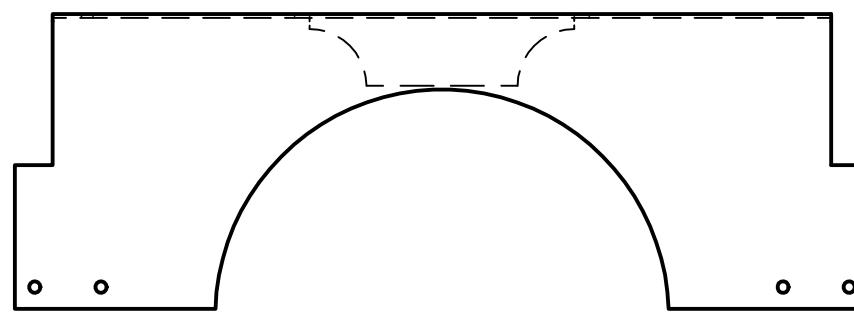
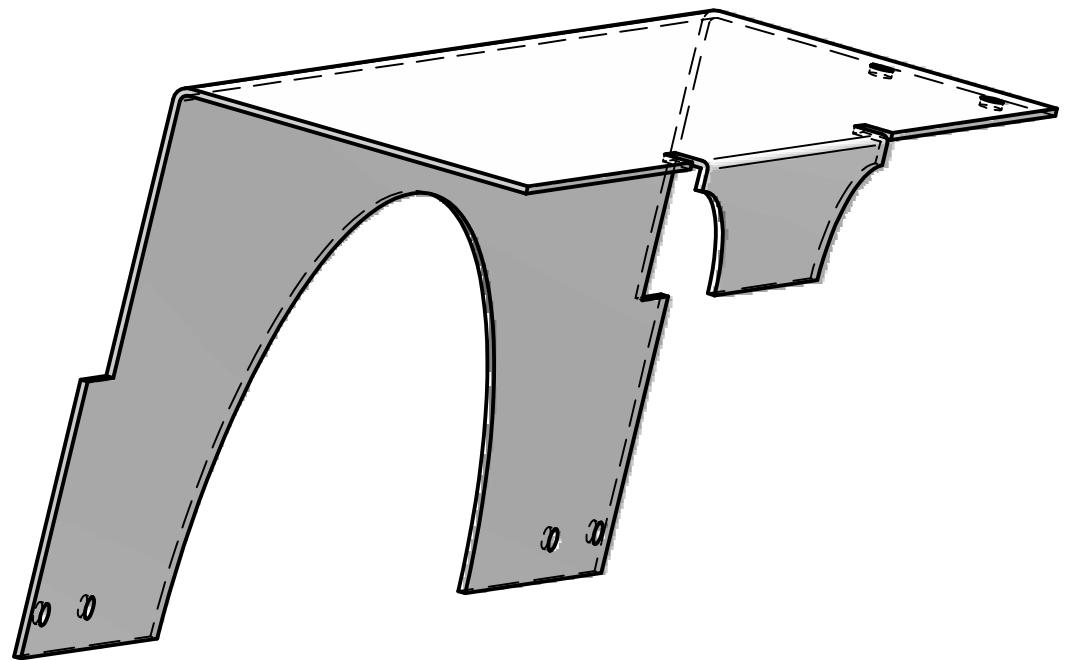
HOJA	NUMERO PLANO	
A4	00-01-22-050	
REV		
0		
ESCALA:	2:1	PESO:
		HOJA 1 DE 1





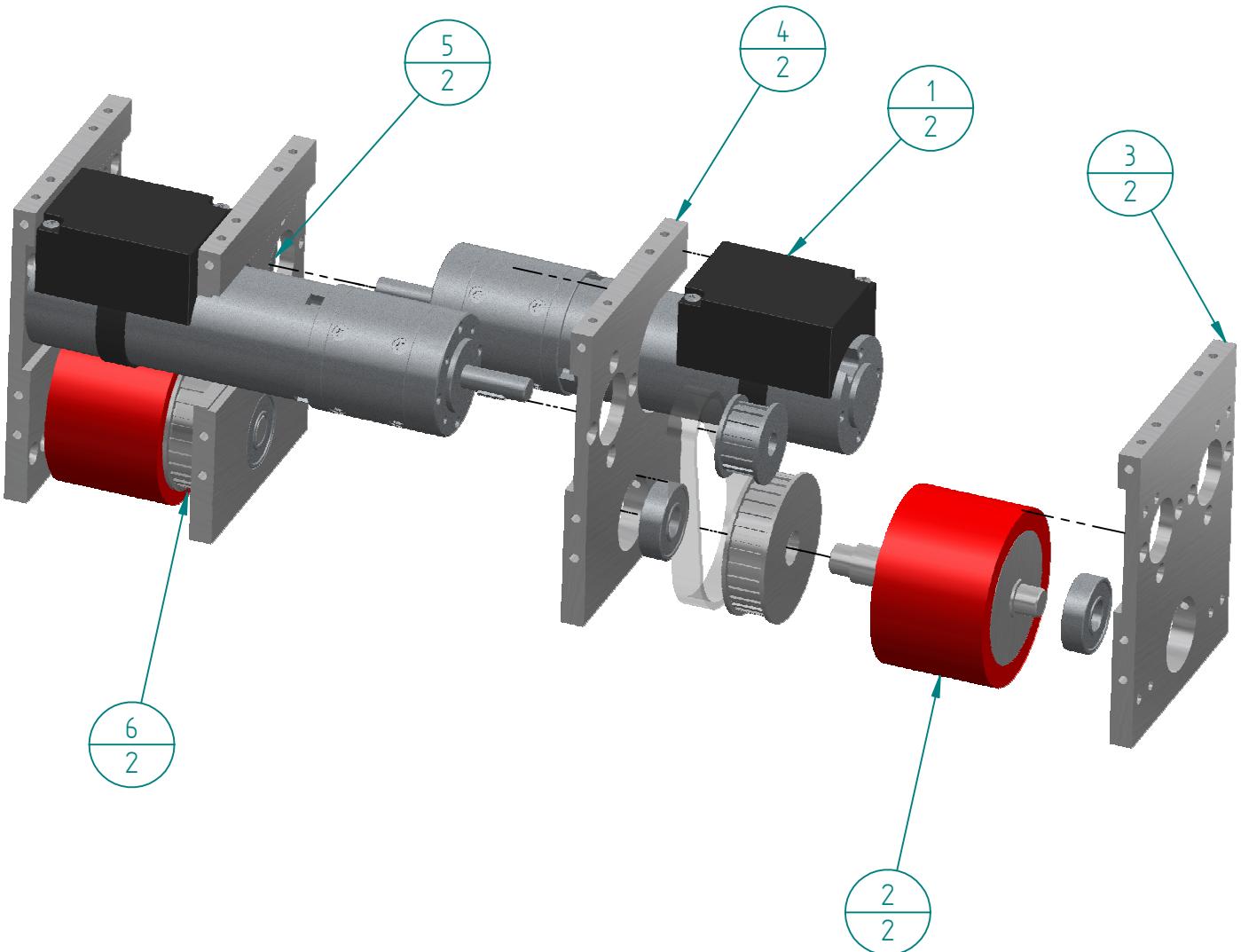
NOMBRE	M.I.G	FECHA	Proyecto Eurobot 2015	
MATERIAL	CHAPA ALUMINIO		TITULO	
ESPESOR	0,5 MM		TRAMPILLA TRASERA INFERIOR	
CANTIDAD	1 S/D + 1 M/C			
DIMENSIONES ACOTADAS EN MILIMETROS	HOJA A3	NUMERO PLANO 00-02-01-013	REV 0	
TOLERANCIA GENERAL ±0,2				
ANGULO ±1°	ESCALA: 1:1	PESO:	HOJA 1 DE 1	





NOMBRE	M.I.G	FECHA	Proyecto Eurobot 2015	
MATERIAL	CHAPA ALUMINIO		TITULO	
ESPESOR	0,5 MM		TRAMPILLA TRASERA SUPERIOR	
CANTIDAD	1 UNIDADES			
DIMENSIONES ACOTADAS EN MILIMETROS		HOJA A3 NUMERO PLANO 00-02-02-017		REV 0
TOLERANCIA GENERAL ±0.2				
ANGULO ±1°		ESCALA: 1:2-1:1 PESO:		HOJA 1 DE 1

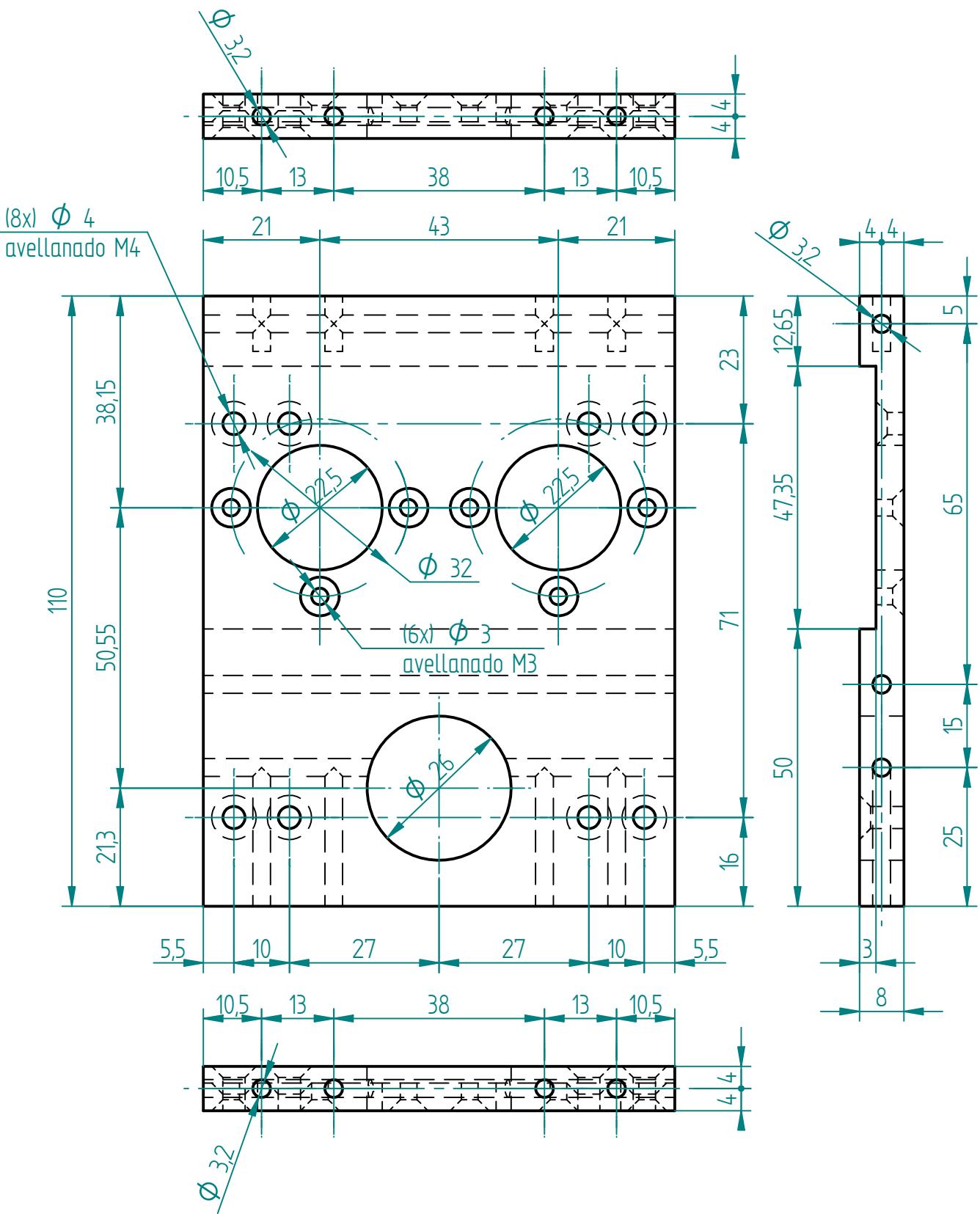




Item	Plano	N	Masa
1	motor dunkermotoren	2	3,115 kg
2	BLOQUE RUEDAS	2	0,545 kg
3	PLACA ENCODER IZQUIERDO	2	0,270 kg
4	PLACA MOTOR IZQUIERDA	2	0,245 kg
5	1.15	2	0,092 kg
6	CORREA TRACCION	2	0,012 kg

DIBUJADO	NOMBRE	FECHA	Proyecto Eurobot 2015		
	M.I.G	12-01-2015			
MATERIAL					
ESPESOR					
CANTIDAD	1 S/C + 1 M/C				
DIMENSIONES ACOTADAS EN MILIMETROS TOLERANCIA GENERAL ±0.2 ANGULO ±1°			HOJA	NUMERO PLANO	REV
			A4	01-00-00-E01	0
			ESCALA:	PESO:	HOJA 1 DE 1





	NOMBRE	FECHA
DIBUJADO	M.I.G	12-01-15
MATERIAL	ALUMINIO 5083	
ESPESOR	8 MM	
CANTIDAD	2 UNIDADES	

Proyecto Eurobot 2015

TITULO

PLACA ENCODER

HOJA

A4

NUMERO PLANO

01-00-01-000

REV

0

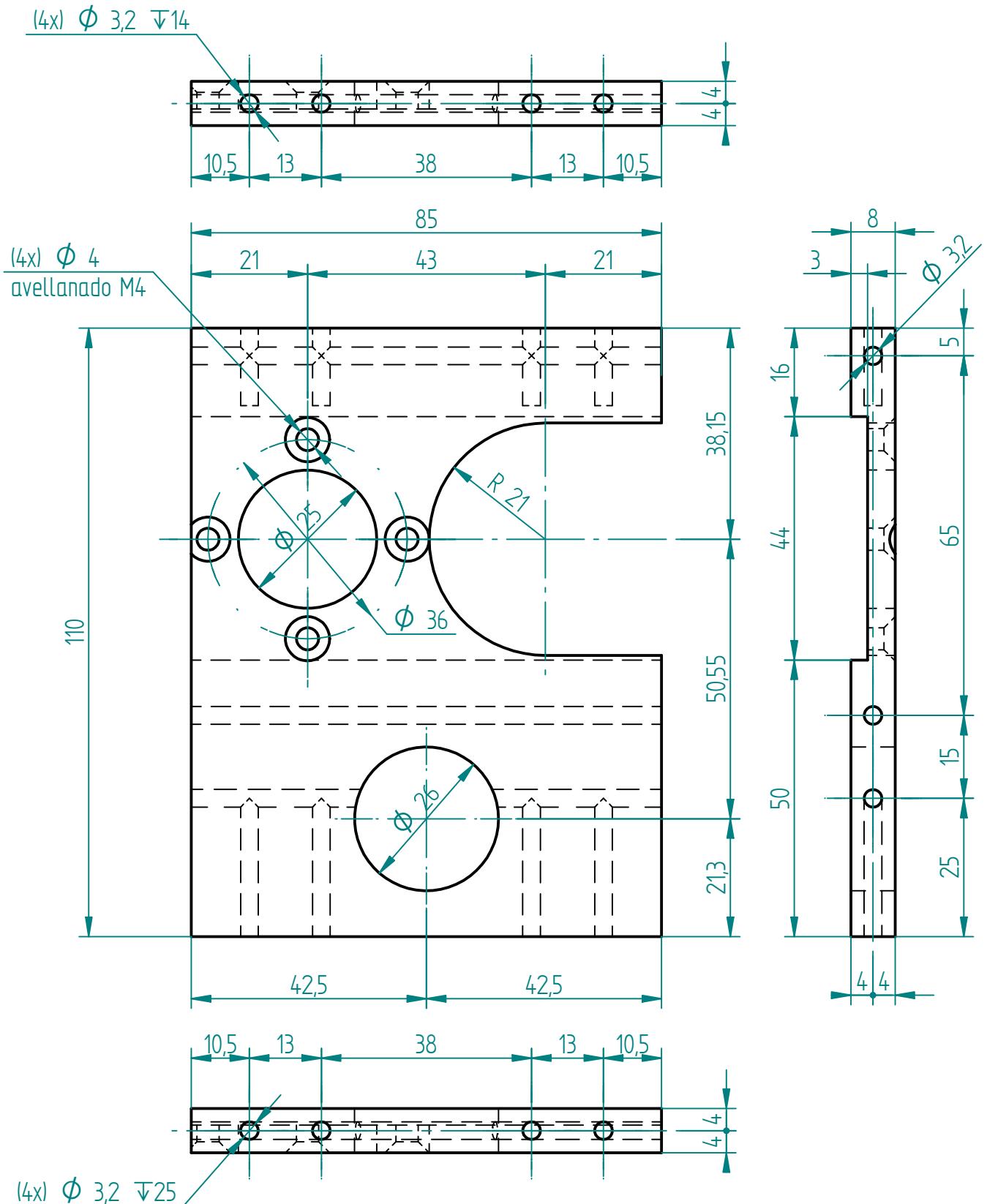
ESCALA:

1:1

PESO:

HOJA 1 DE 1





	NOMBRE	FECHA
DIBUJADO	M.I.G	12-01-15
MATERIAL	ALUMINIO 5083	
ESPESOR	8 MM	
CANTIDAD	2 UNIDADES	
DIMENSIONES ACOTADAS EN MILIMETROS		
TOLERANCIA GENERAL $\pm 0.2$		
ANGULO $\pm 1^\circ$		

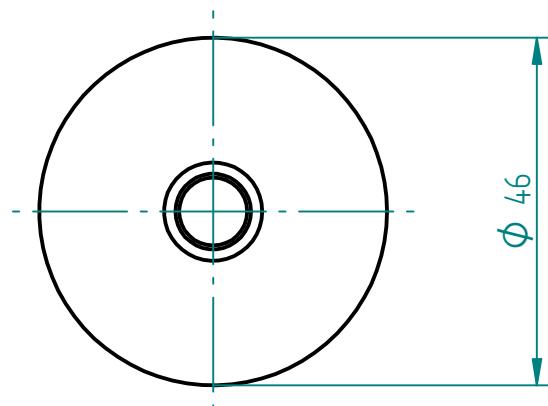
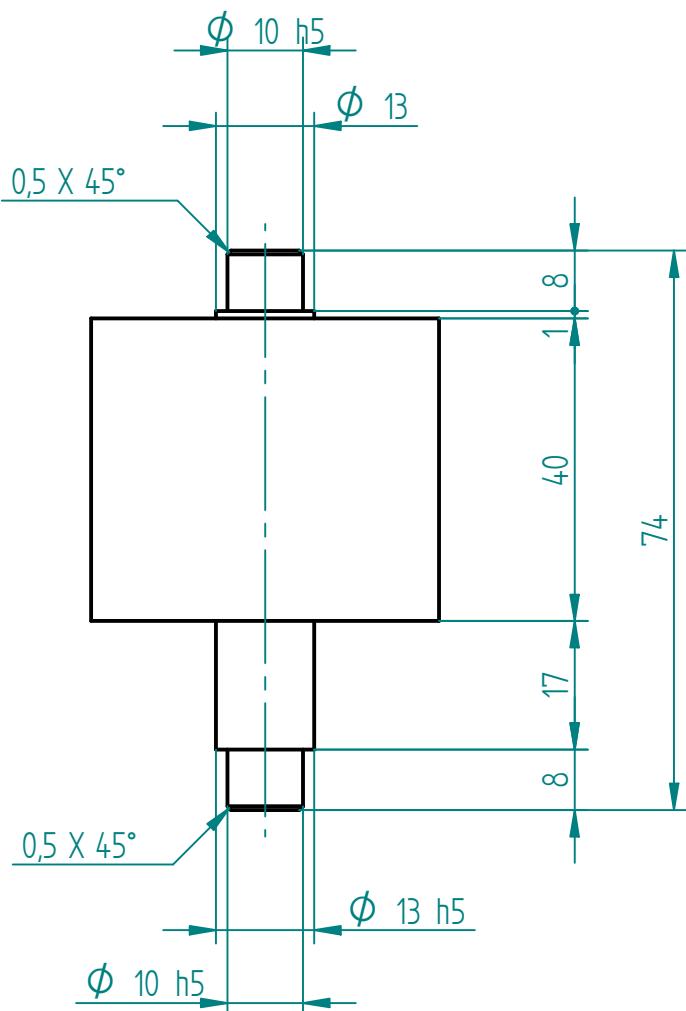
Proyecto Eurobot 2015

TITULO

PLACA MOTOR

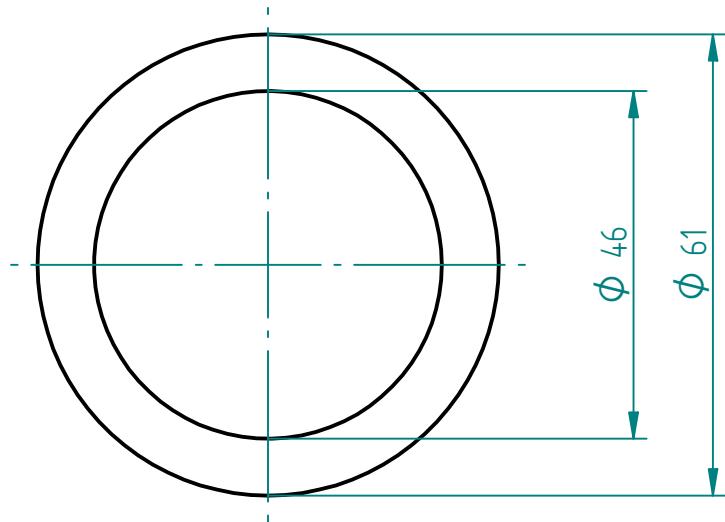
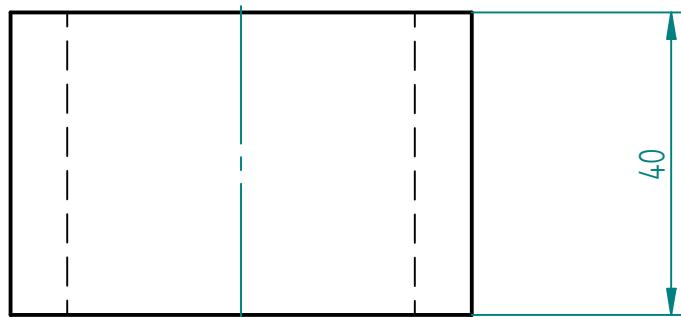
HOJA	NUMERO PLANO	REV
A4	01-00-02-000	0
ESCALA: 1:1	PESO:	HOJA 1 DE 1





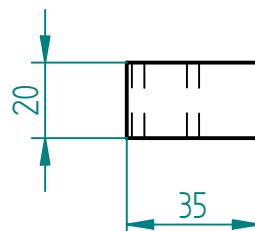
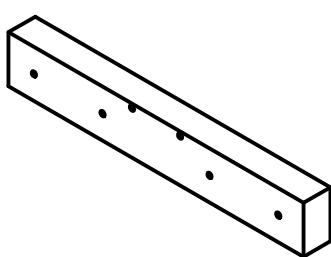
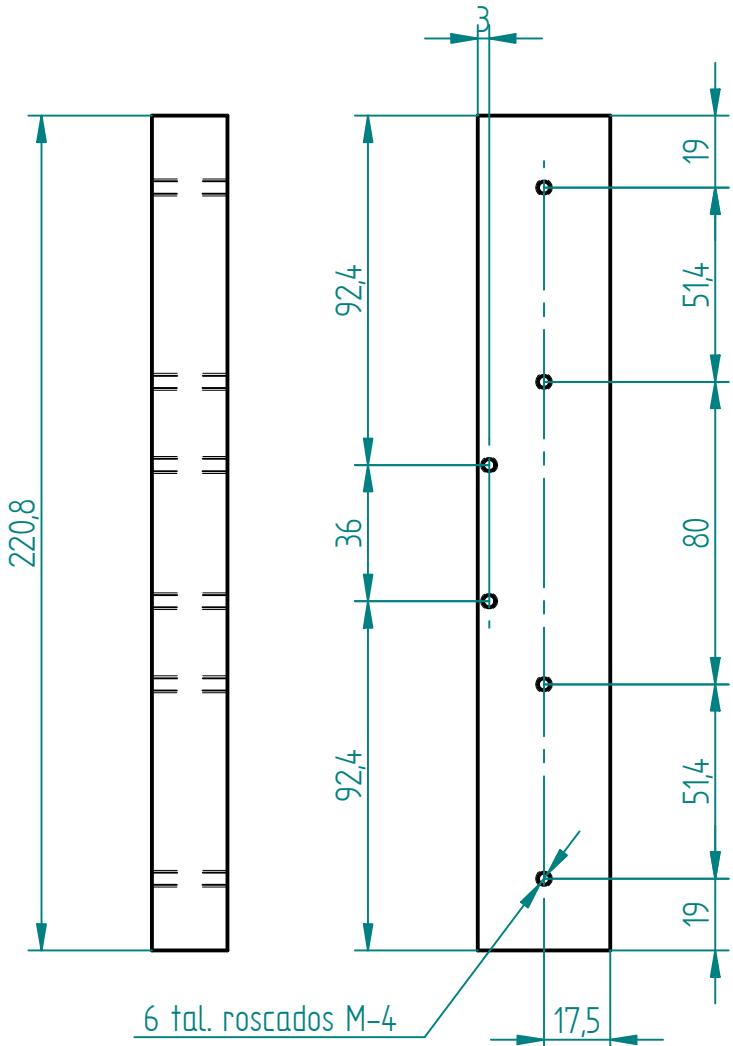
NOMBRE		FECHA	Proyecto Eurobot 2015		
DIBUJADO	M.I.G	12-01-15			
MATERIAL	ALUMINIO DURAL			TITULO	
ESPESOR				LLANTA RUEDA	
CANTIDAD	2 UNIDADES				
DIMENSIONES ACOTADAS EN MILIMETROS			HOJA	NUMERO PLANO	REV
TOLERANCIA GENERAL ±0.2			A4	01-00-03-000	0
ANGULO ±1°			ESCALA: 1:1	PESO:	HOJA 1 DE 1





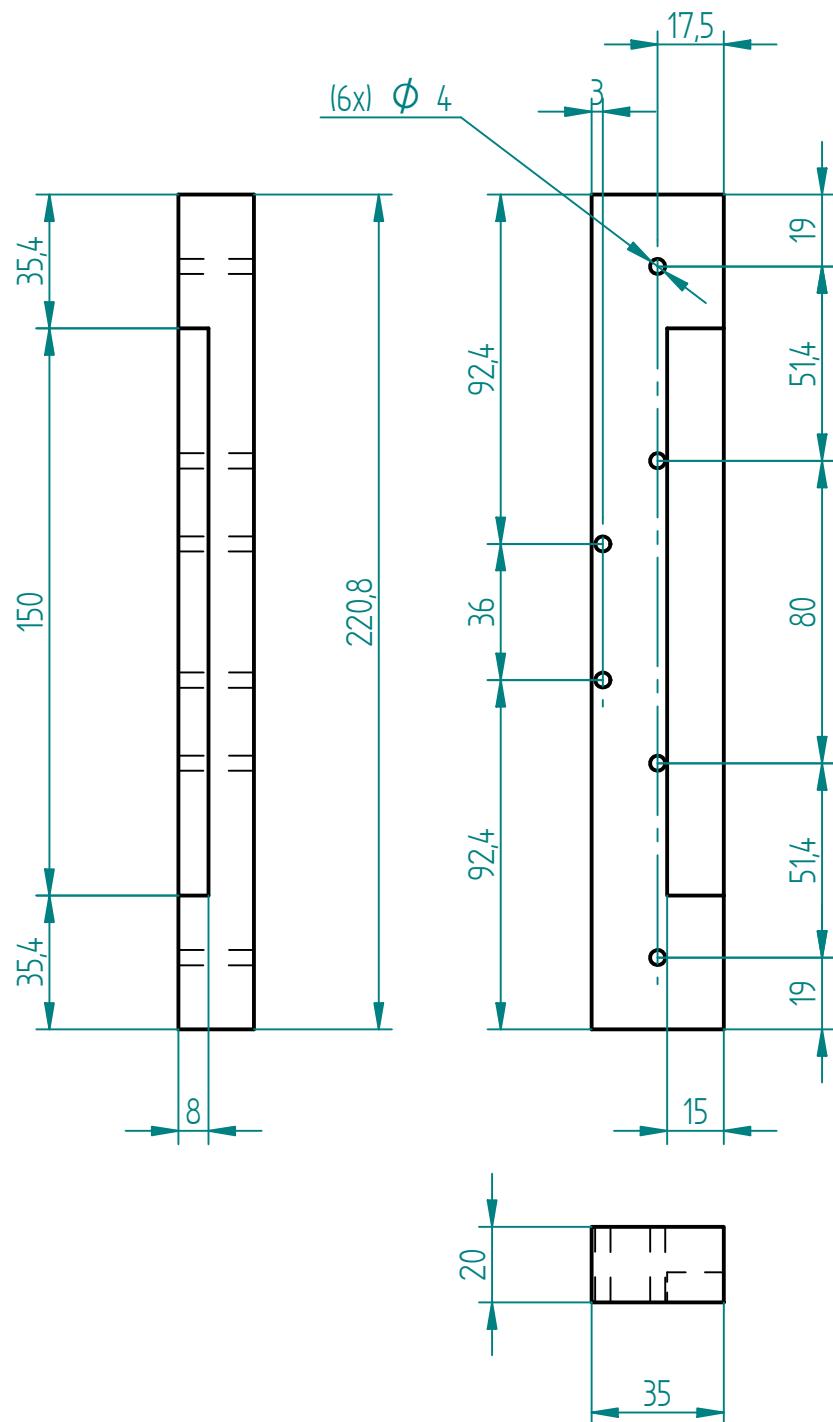
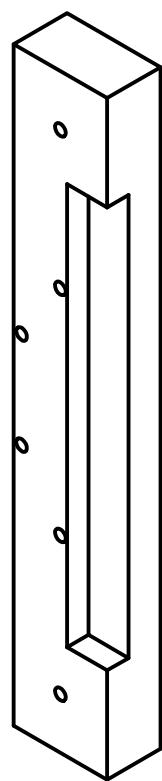
NOMBRE	FECHA	Proyecto Eurobot 2015		
DIBUJADO	M.I.G	12-01-15		
MATERIAL	TUBO FOAM	TITULO		
ESPESOR	7,5 MM	RUEDA FOAM ANCHO 40		
CANTIDAD	2 UNIDADES			
DIMENSIONES ACOTADAS EN MILIMETROS		HOJA	NUMERO PLANO	REV
TOLERANCIA GENERAL ±0.2		A4	01-00-04-000	0
ANGULO ±1°		ESCALA:	1:1	PESO:
				HOJA 1 DE 1





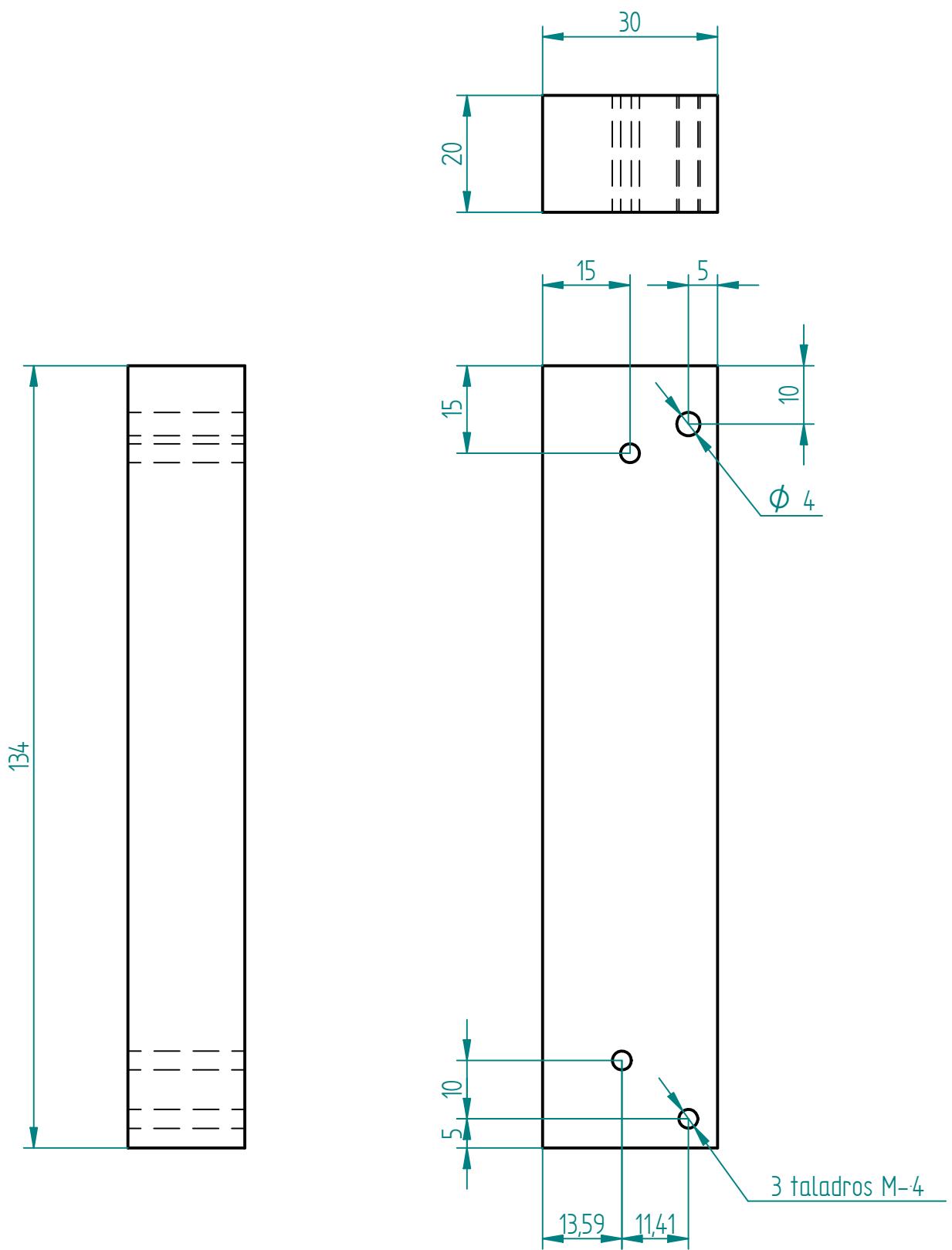
NOMBRE	M.I.G	FECHA	Proyecto Eurobot 2015		
DIBUJADO					
MATERIAL	ACERO		TITULO		
ESPESOR			CONTRAPESO TRASERO		
CANTIDAD	1 UNIDADES				
DIMENSIONES ACOTADAS EN MILIMETROS			HOJA	NUMERO PLANO	REV
TOLERANCIA GENERAL ±0.2			A4	02-00-01-006	0
ANGULO ±1°			ESCALA: 1:2	PESO:	HOJA 1 DE 1





DIBUJADO	NOMBRE	FECHA	Proyecto Eurobot 2015		
MATERIAL	ACERO				
ESPESOR				TITULO	
CANTIDAD	1 UNIDADES			CONTRAPESO TRASERO SUPERIOR	
DIMENSIONES ACOTADAS EN MILIMETROS TOLERANCIA GENERAL $\pm 0.2$ ANGULO $\pm 1^\circ$			HOJA	NUMERO PLANO	REV
			A4	02-00-02-006	0
			ESCALA: 1:2		PESO:
			HOJA 1 DE 1		





NOMBRE	M.I.G	FECHA
DIBUJADO	M.I.G	21-02-15
MATERIAL	ACERO F-100	
ESPESOR		
CANTIDAD	2 UNIDADES	
DIMENSIONES ACOTADAS EN MILIMETROS		
TOLERANCIA GENERAL $\pm 0.2$		
ANGULO $\pm 1^\circ$		

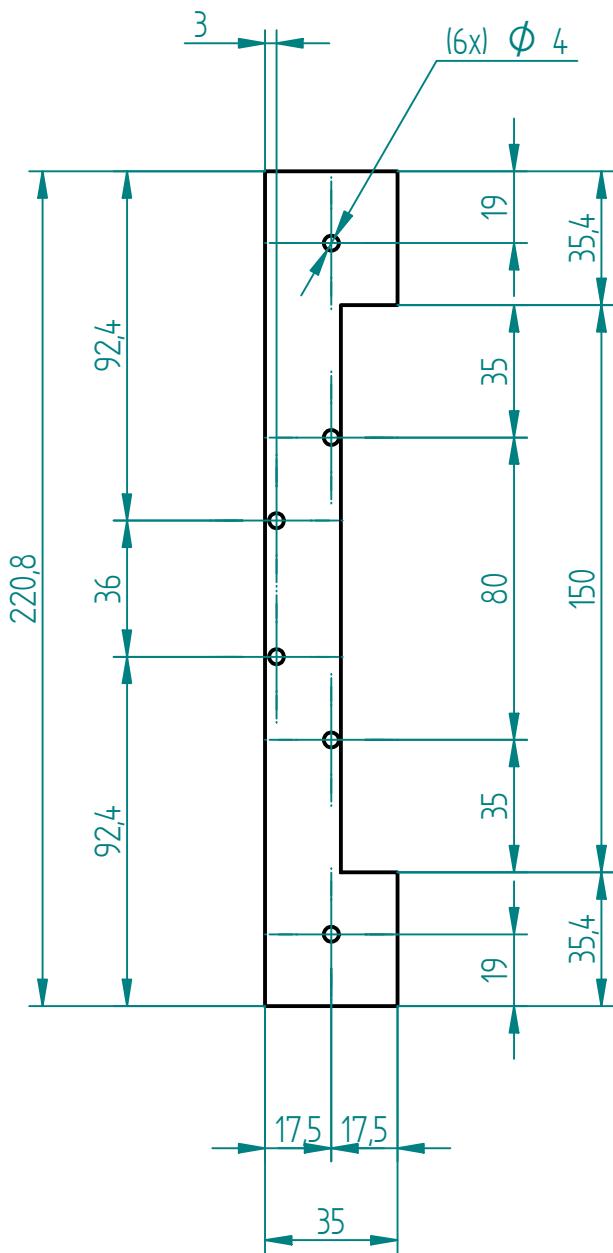
Proyecto Eurobot 2015

TITULO

CONTRAPESO DELANTERO

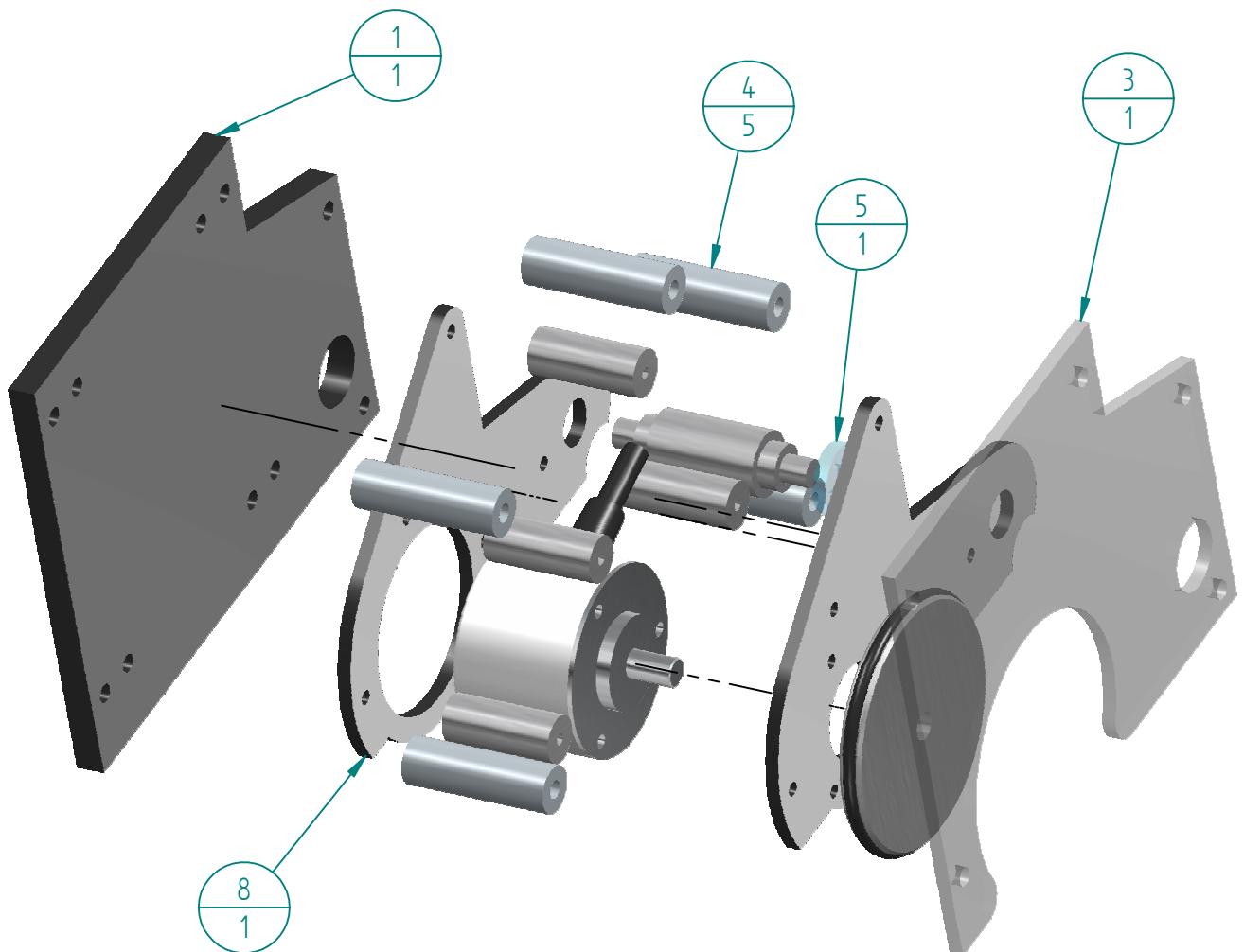
HOJA	NUMERO PLANO	REV
A4	02-00-03-005	0
ESCALA: 1:1	PESO:	HOJA 1 DE 1





NOMBRE	M.I.G	FECHA	Proyecto Eurobot 2015		
DIBUJADO		28-02-15			
MATERIAL	TRESPA				
ESPESOR	3 MM				
CANTIDAD	1 UNIDADES		TITULO		
DIMENSIONES ACOTADAS EN MILIMETROS			SUPLEMENTO TRASERO		
TOLERANCIA GENERAL ±0.2			A4	NUMERO PLANO	REV
ANGULO ±1°				02-00-04-204	0
			ESCALA: 1:2	PESO:	HOJA 1 DE 1

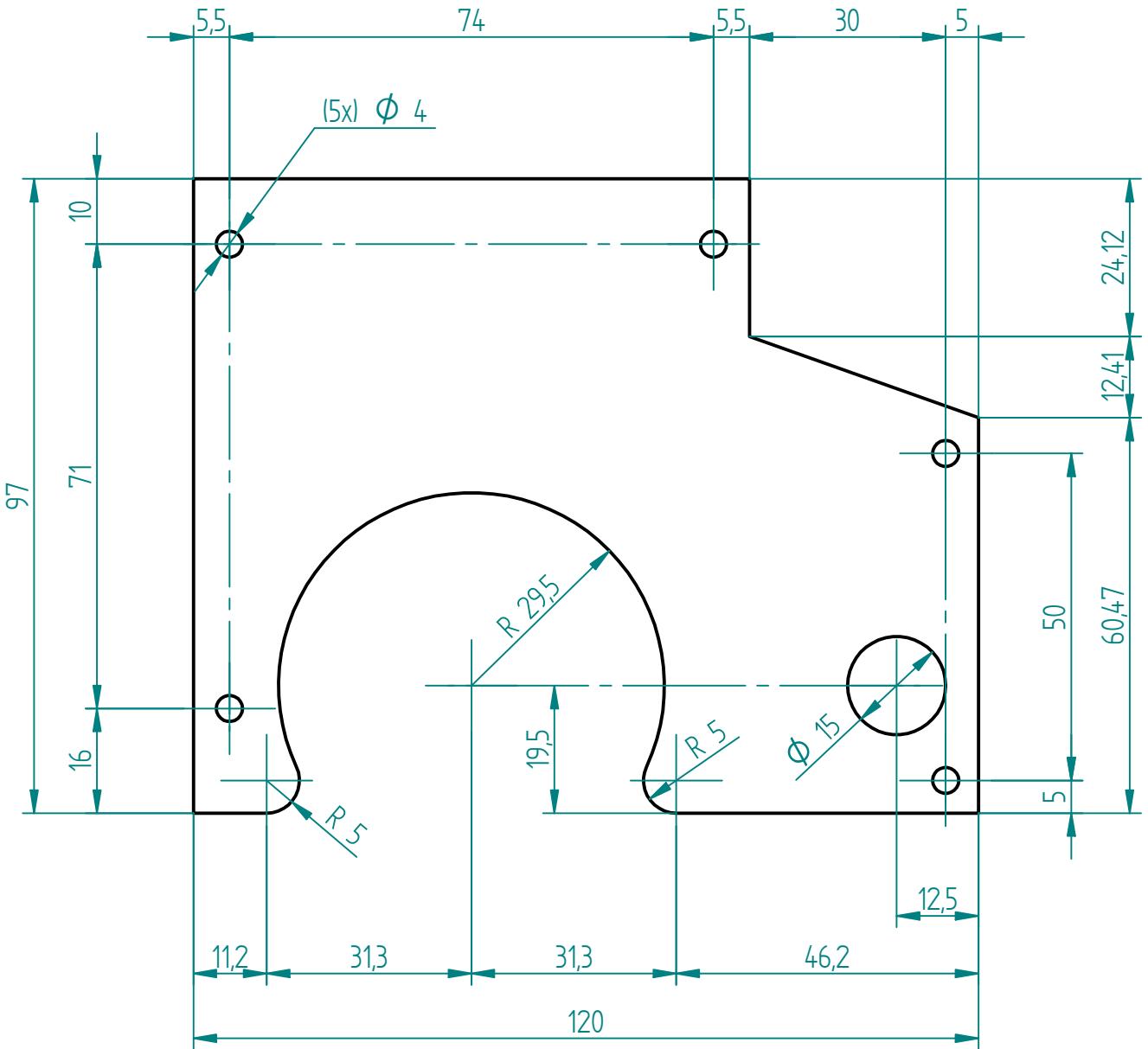




Item	Plano	N	Masa
1	Bisgra Interior Balancin	1	0,080 kg
3	Bisgra Exterior Balancin	1	0,031 kg
4	Separadores Bisagra	5	0,000 kg
5	Rodamiento Balancin	1	0,000 kg
6*	Brazo Exterior Balancin	1	0,017 kg
7*	Encoder Hengstler RI41	1	0,098 kg
8	Brazo Interior Balancin	1	0,014 kg
9*	Separador Brazos Encoder	4	0,019 kg
10*	Eje Brazos Encoder	1	0,013 kg
11*	RUEDA ENCODER	1	0,030 kg

DIBUJADO	NOMBRE	FECHA	Proyecto Eurobot 2015		
	M.I.G	25-01-2015			
MATERIAL	.				
ESPESOR	.				
CANTIDAD	1 S/D + 1 M/C				
DIMENSIONES ACOTADAS EN MILIMETROS TOLERANCIA GENERAL ±0.2 ANGULO ±1°			HOJA A4	NUMERO PLANO 03-00-00-E01	REV 0
			ESCALA:	PESO:	HOJA 1 DE 1



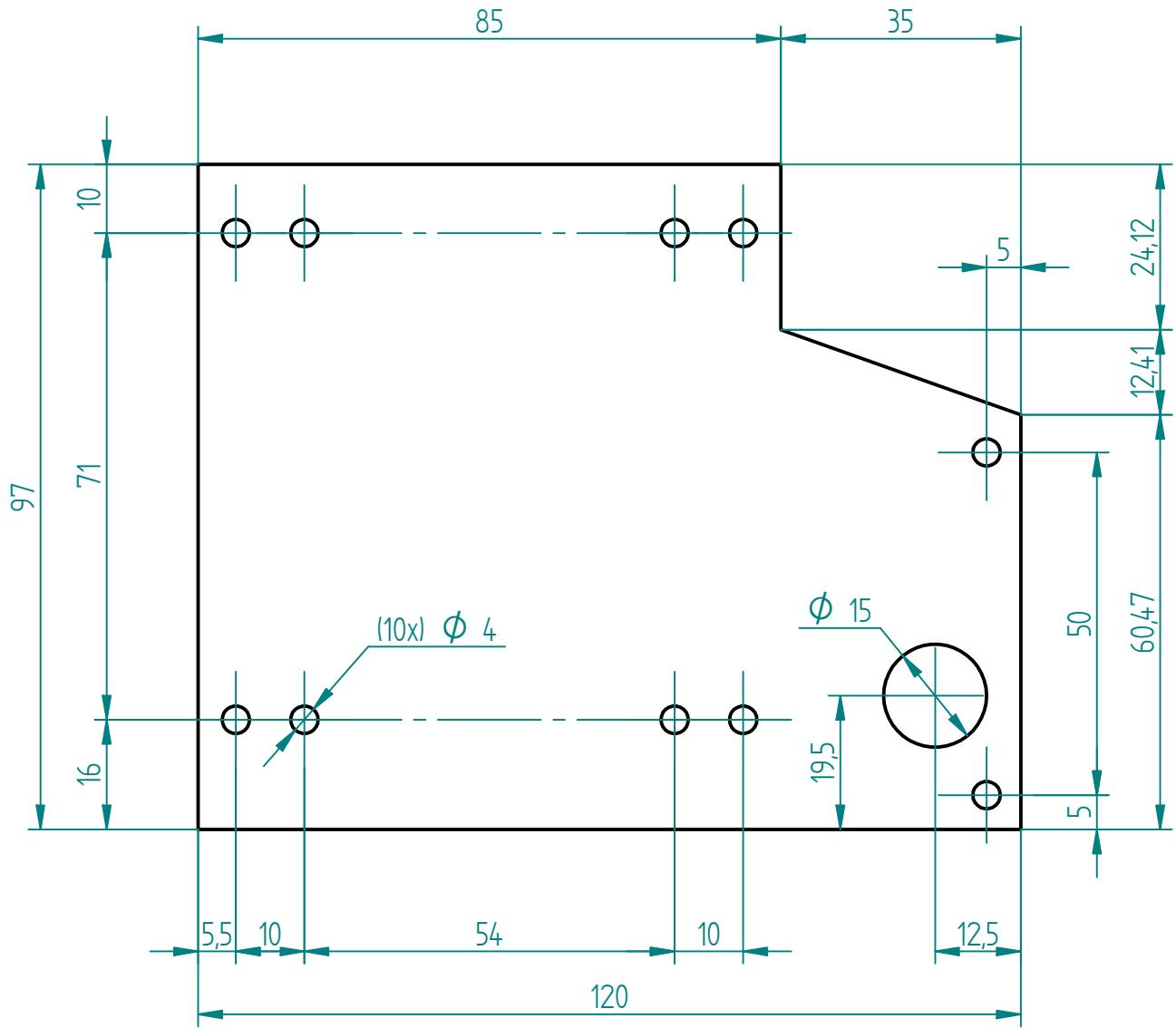


	NOMBRE	FECHA
DIBUJADO	M.I.G	25-01-15
MATERIAL	TRESPA	
ESPESOR	3 MM	
CANTIDAD	2 UNIDADES	
DIMENSIONES ACOTADAS EN MILIMETROS		
TOLERANCIA GENERAL ±0.2		
ANGULO ±1°		

## Proyecto Eurobot 2015

TITULO	BISAGRA EXTERIOR BALANCIN ENCODER	
HOJA	NUMERO PLANO	REV
A4	03-00-01-030	0
ESCALA: 1:1	PESO:	HOJA 1 DE 1



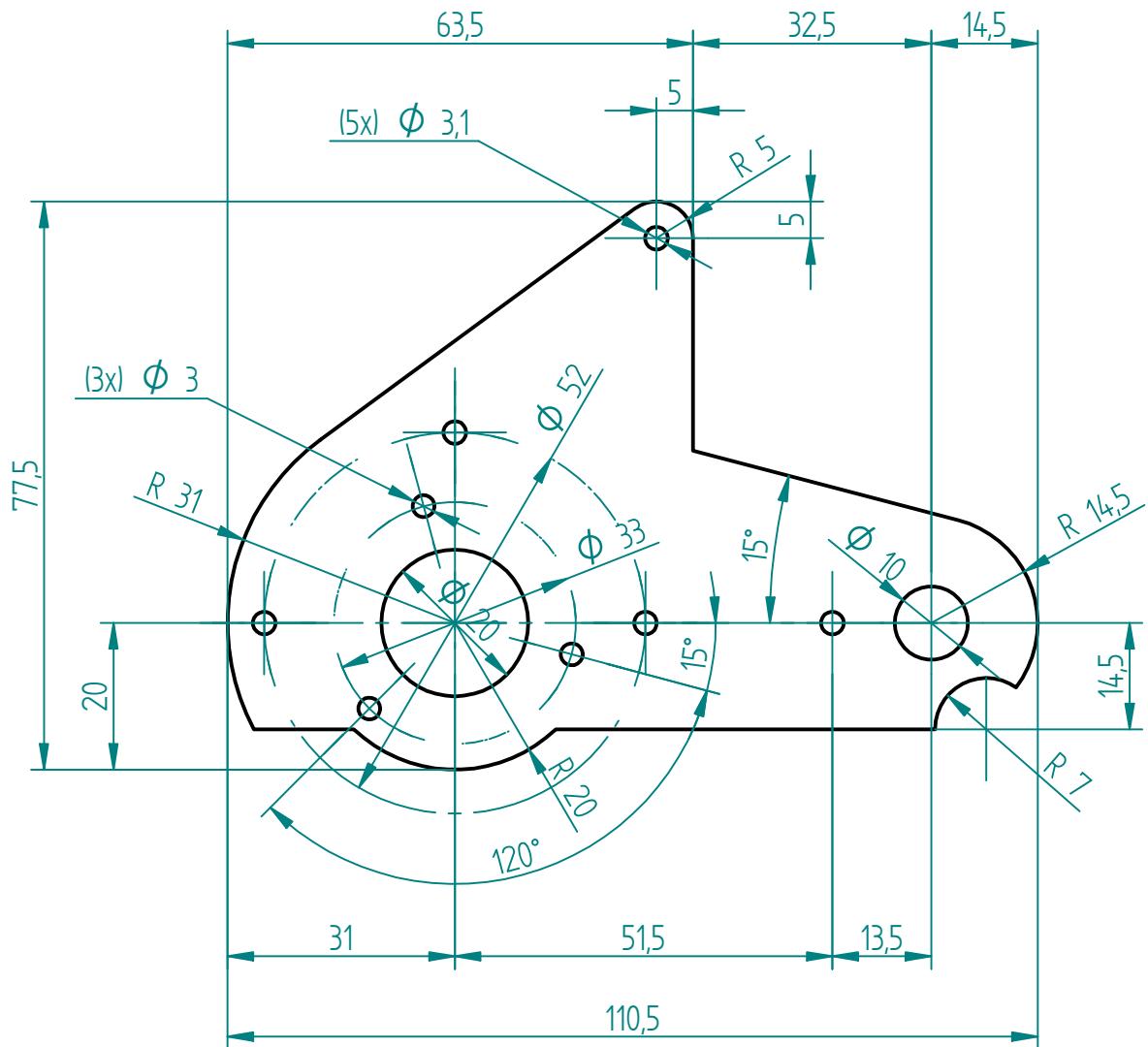


	NOMBRE	FECHA
DIBUJADO	M.I.G	25-01-15
MATERIAL	TRESPA	
ESPESOR	3 MM	
CANTIDAD	2 UNIDADES	
DIMENSIONES ACOTADAS EN MILIMETROS		
TOLERANCIA GENERAL $\pm 0.2$		
ANGULO $\pm 1^\circ$		

Proyecto Eurobot 2015

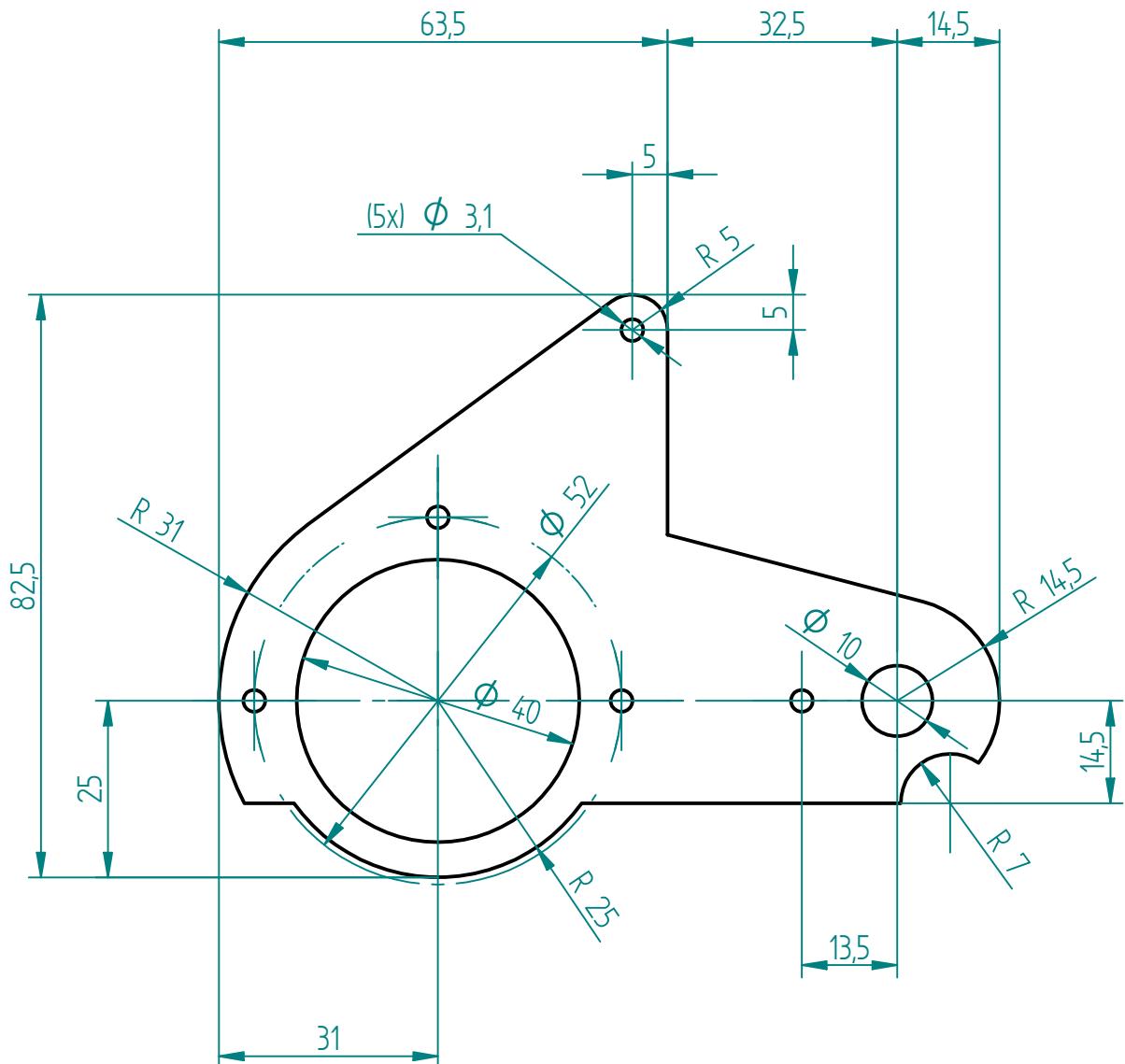
TITULO	BISAGRA INTERIOR BALANCIN ENCODER	
HOJA	NUMERO PLANO	REV
A4	03-00-02-031	0
ESCALA:	PESO:	HOJA 1 DE 1





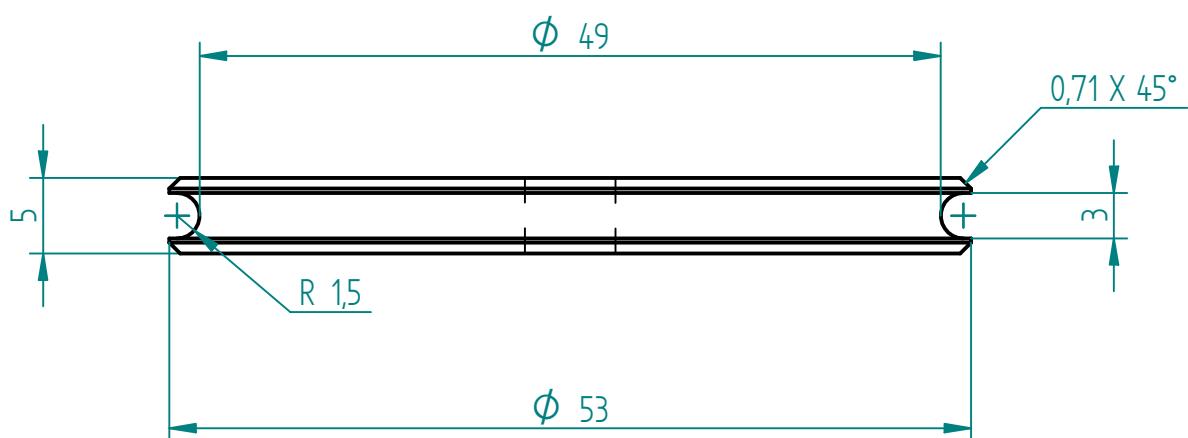
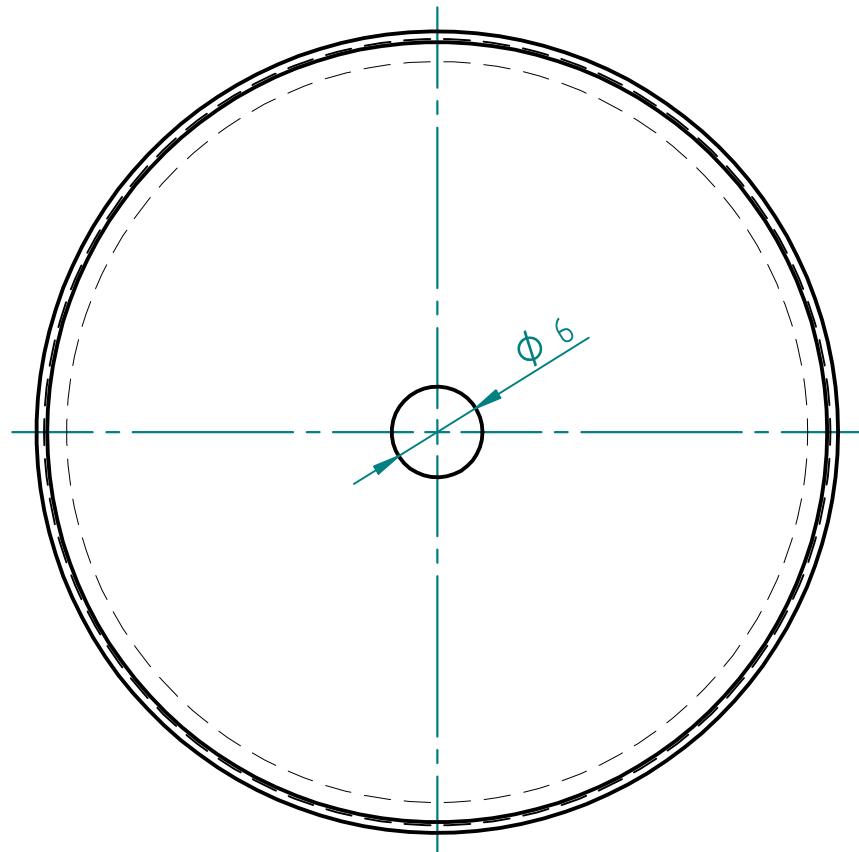
NOMBRE	M.I.G	FECHA	Proyecto Eurobot 2015		
DIBUJADO					
MATERIAL	TRESPA		TITULO		
ESPESOR	3 MM		BRAZO EXTERIOR BALANCIN ENCODER		
CANTIDAD	2 UNIDADES				
DIMENSIONES ACOTADAS EN MILIMETROS TOLERANCIA GENERAL $\pm 0.2$ ANGULO $\pm 1^\circ$			HOJA	NUMERO PLANO	REV
			A4	03-00-03-032	0
			ESCALA:	PESO:	HOJA 1 DE 1





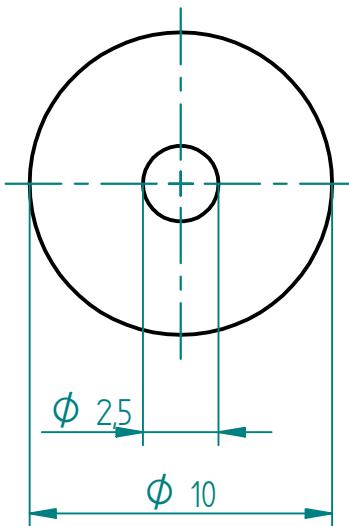
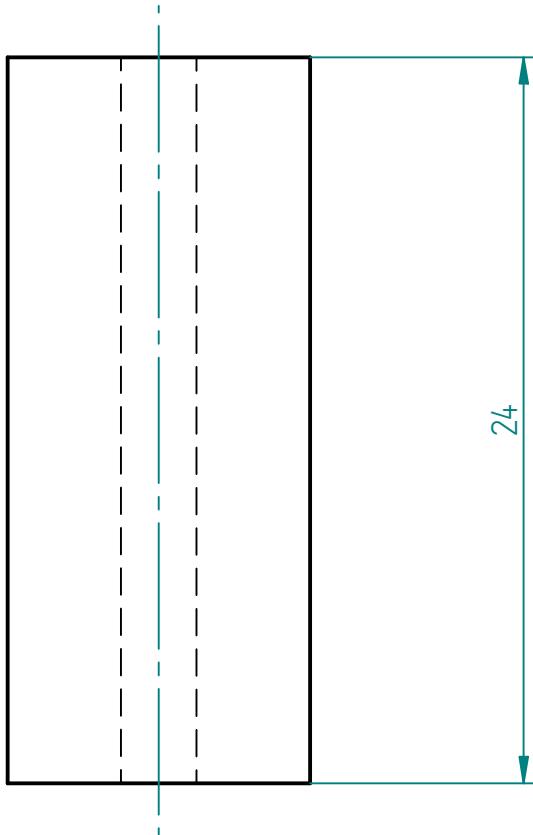
NOMBRE		FECHA	Proyecto Eurobot 2015		
DIBUJADO	M.I.G	25-01-15			
MATERIAL	TRESPA				
ESPESOR	3 MM				
CANTIDAD	2 UNIDADES				
DIMENSIONES ACOTADAS EN MILIMETROS TOLERANCIA GENERAL ±0.2 ANGULO ±1°			HOJA A4	NUMERO PLANO 03-00-04-033	REV 0
			ESCALA: 1:1	PESO:	HOJA 1 DE 1





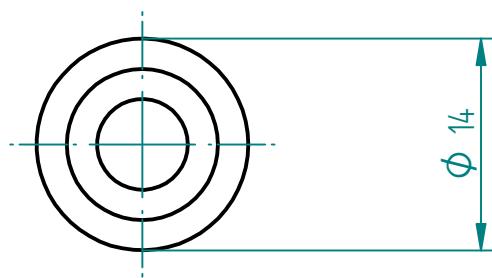
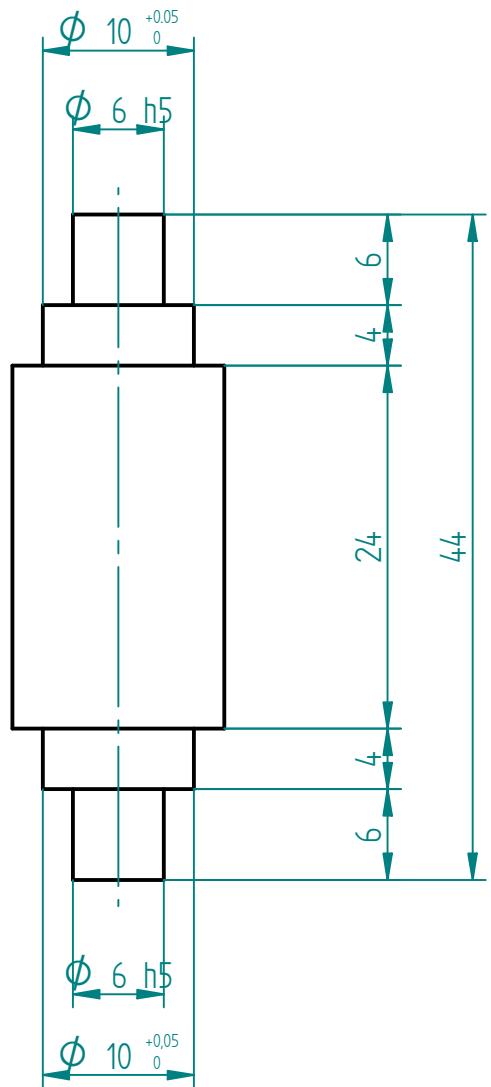
NOMBRE	M.I.G	FECHA	Proyecto Eurobot 2015		
DIBUJADO		25-01-15			
MATERIAL	ALUMINIO DURAL			TITULO	
ESPESOR	5 MM			ROEDA BALANCIN ENCODER	
CANTIDAD	2 UNIDADES				
DIMENSIONES ACOTADAS EN MILIMETROS			HOJA	NUMERO PLANO	REV
TOLERANCIA GENERAL ±0.2			A4	03-00-05-000	0
ANGULO ±1°			ESCALA:	PESO:	HOJA 1 DE 1





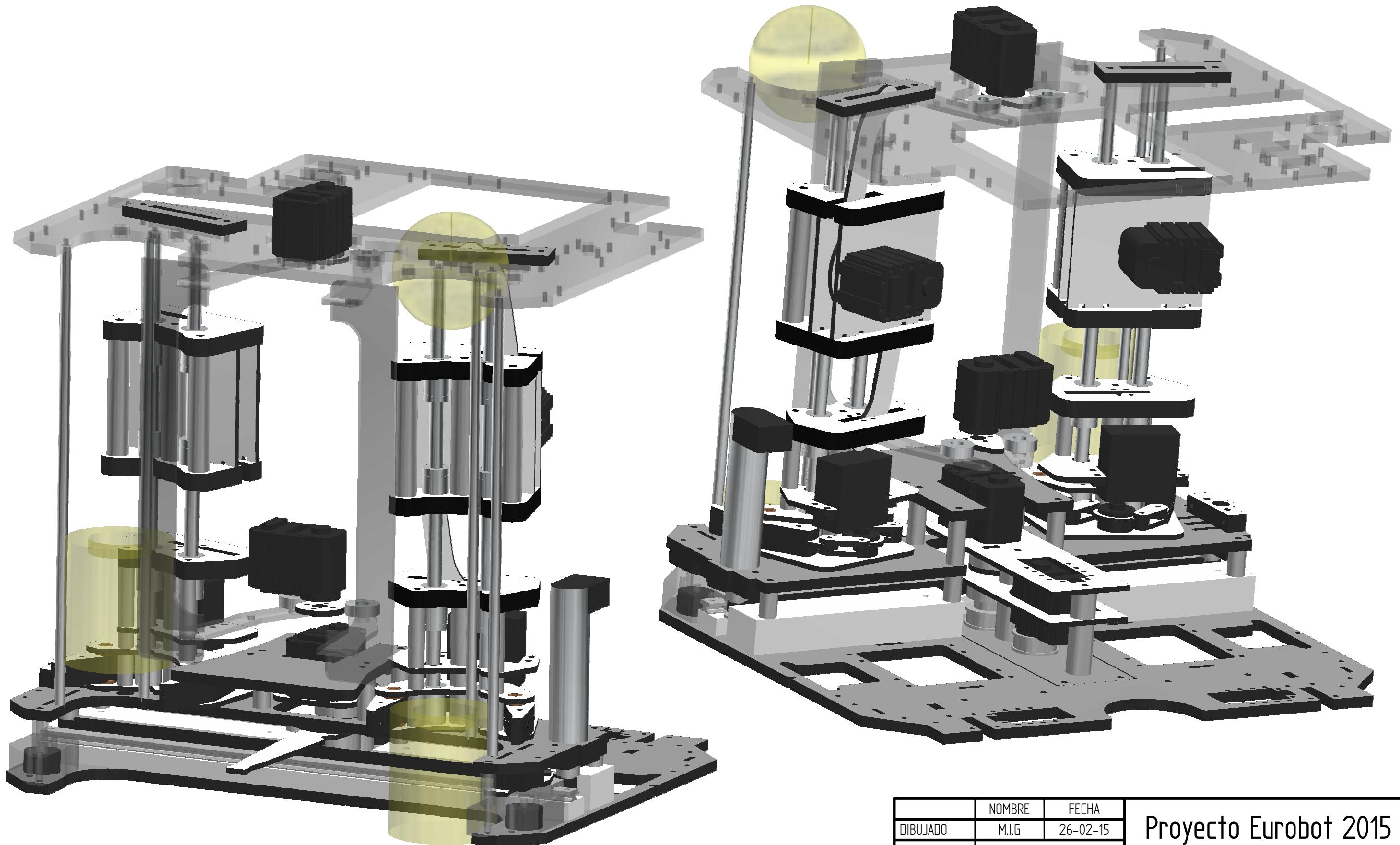
NOMBRE	FECHA	Proyecto Eurobot 2015		
DIBUJADO	M.I.G	25-01-15		
MATERIAL	ALUMINIO DURAL	TITULO		
ESPESOR	MM	SEPARADOR BRAZOS ENCODER		
CANTIDAD	8 UNIDADES			
DIMENSIONES ACOTADAS EN MILIMETROS		HOJA	NUMERO PLANO	REV
TOLERANCIA GENERAL $\pm 0.2$		A4	03-00-06-065	0
ANGULO $\pm 1^\circ$		ESCALA:	4:1	PESO:
				HOJA 1 DE 1





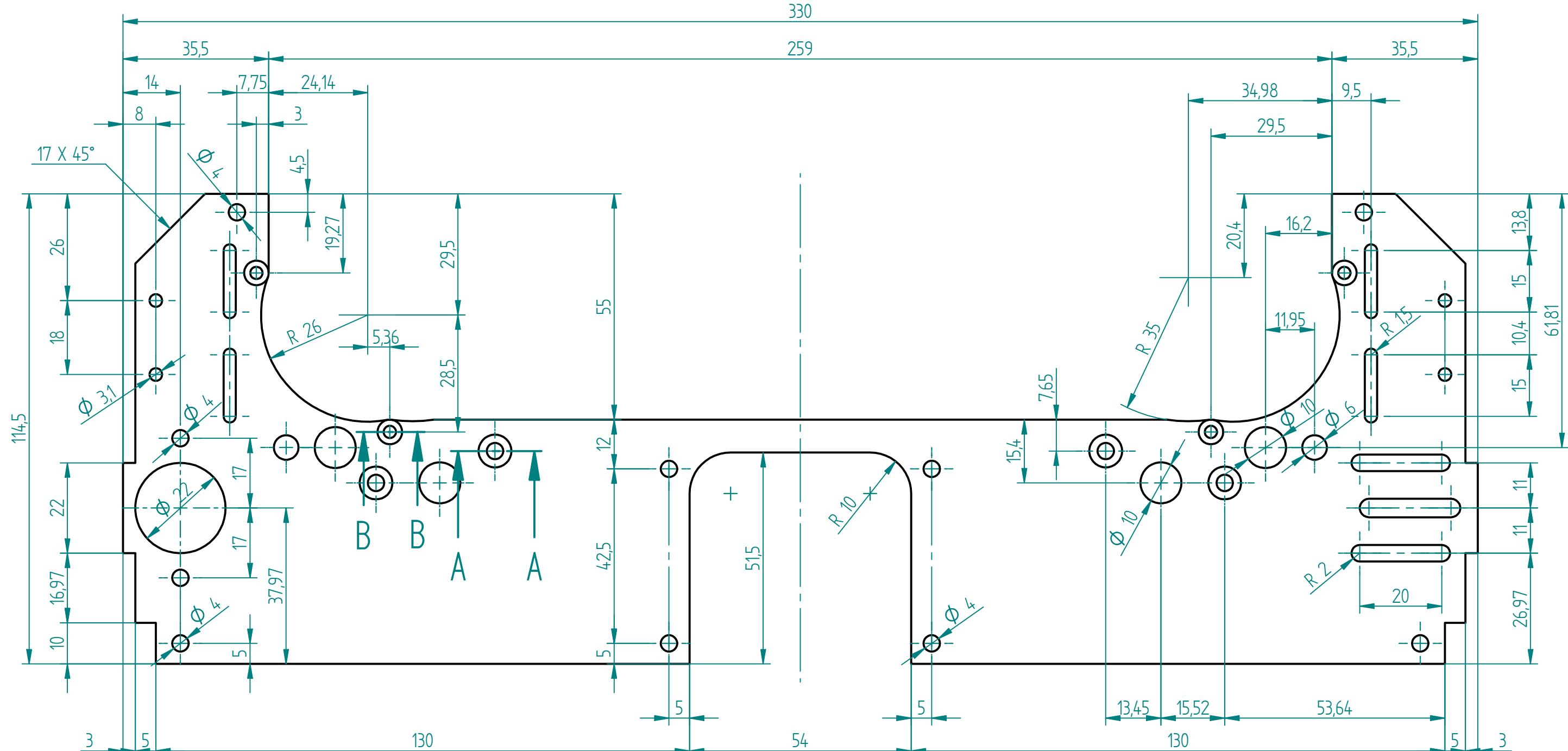
NOMBRE	M.I.G	FECHA	Proyecto Eurobot 2015		
DIBUJADO		25-01-15			
MATERIAL	ALUMINIO DURAL			TITULO	
ESPESOR	MM			EJE BRAZOS ENCODER	
CANTIDAD	2 UNIDADES				
DIMENSIONES ACOTADAS EN MILIMETROS TOLERANCIA GENERAL ±0.2 ANGULO ±1°			HOJA A4	NUMERO PLANO 03-00-07-066	REV 0
			ESCALA: 2:1	PESO:	HOJA 1 DE 1





NOMBRE	FECHA	Proyecto Eurobot 2015	
DIBUJADO	M.I.G	26-02-15	
MATERIAL	.		
ESPESOR	.		
CANTIDAD	.		
TITULO		CONJUNTO COMPLETO SISTEMA STANDS	
DIMENSIONES ACOTADAS EN MILIMETROS TOLERANCIA GENERAL ±0.2 ANGULO ±1°		HOJA A3	NUMERO PLANO 04-00-00-E01
REV 0		ESCALA:	PESO:
		HOJA 1 DE 1	



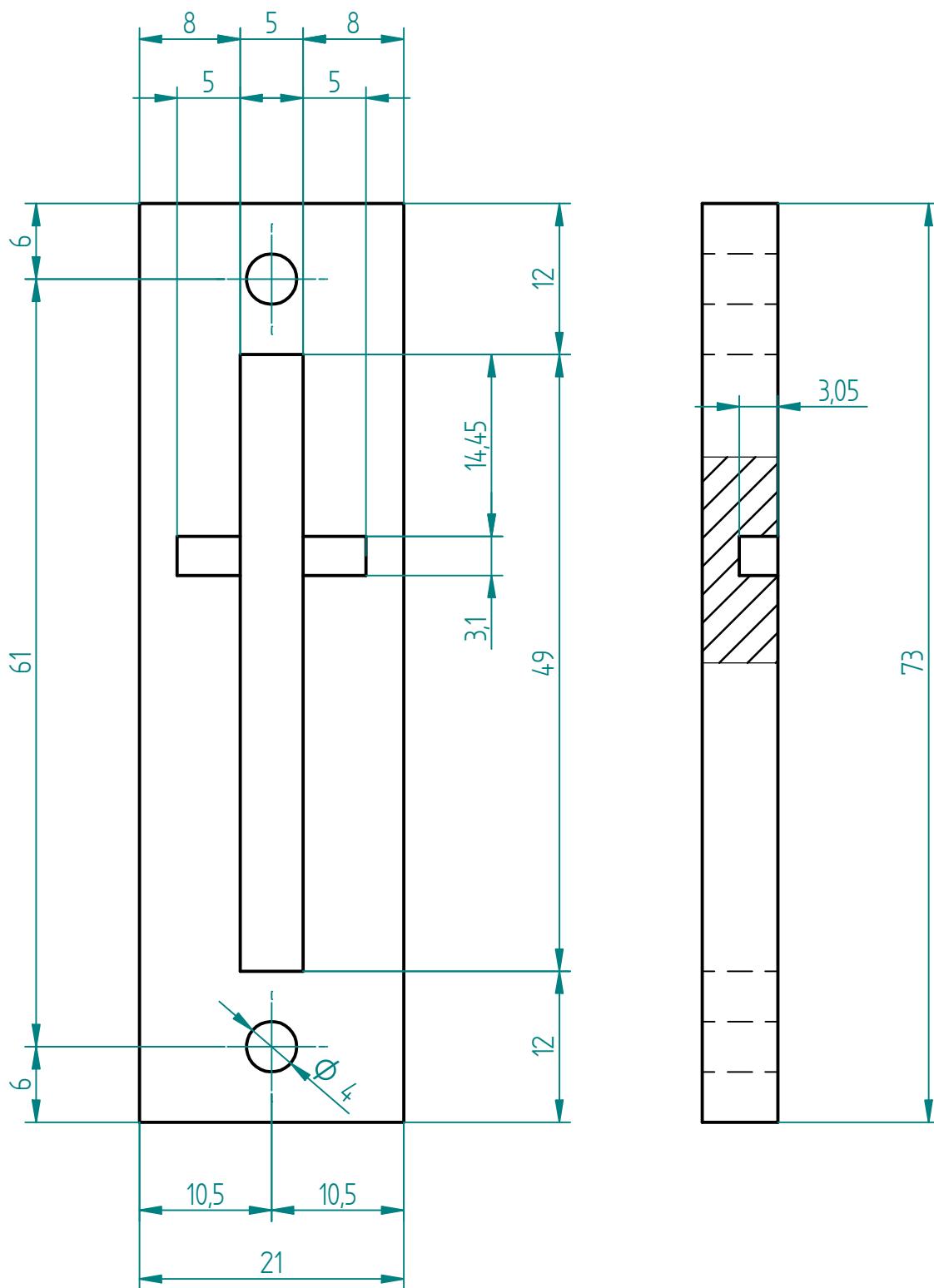


## SECTION B-B

# SECTION A-A

	NOMBRE	FECHA	Proyecto Eurobot 2015		
DIBUJADO	M.I.G	18-01-15			
MATERIAL	TRESPA				
ESPESOR	6 MM				
CANTIDAD	1 UNIDADES				
DIMENSIONES ACOTADAS EN MILIMETROS TOLERANCIA GENERAL ±0.2		HOJA	NUMERO PLANO		REV
		A3	04-00-01-009		0
ANGULO ±1°		ESCALA: 1:1	PESO:	HOJA 1 DE 1	



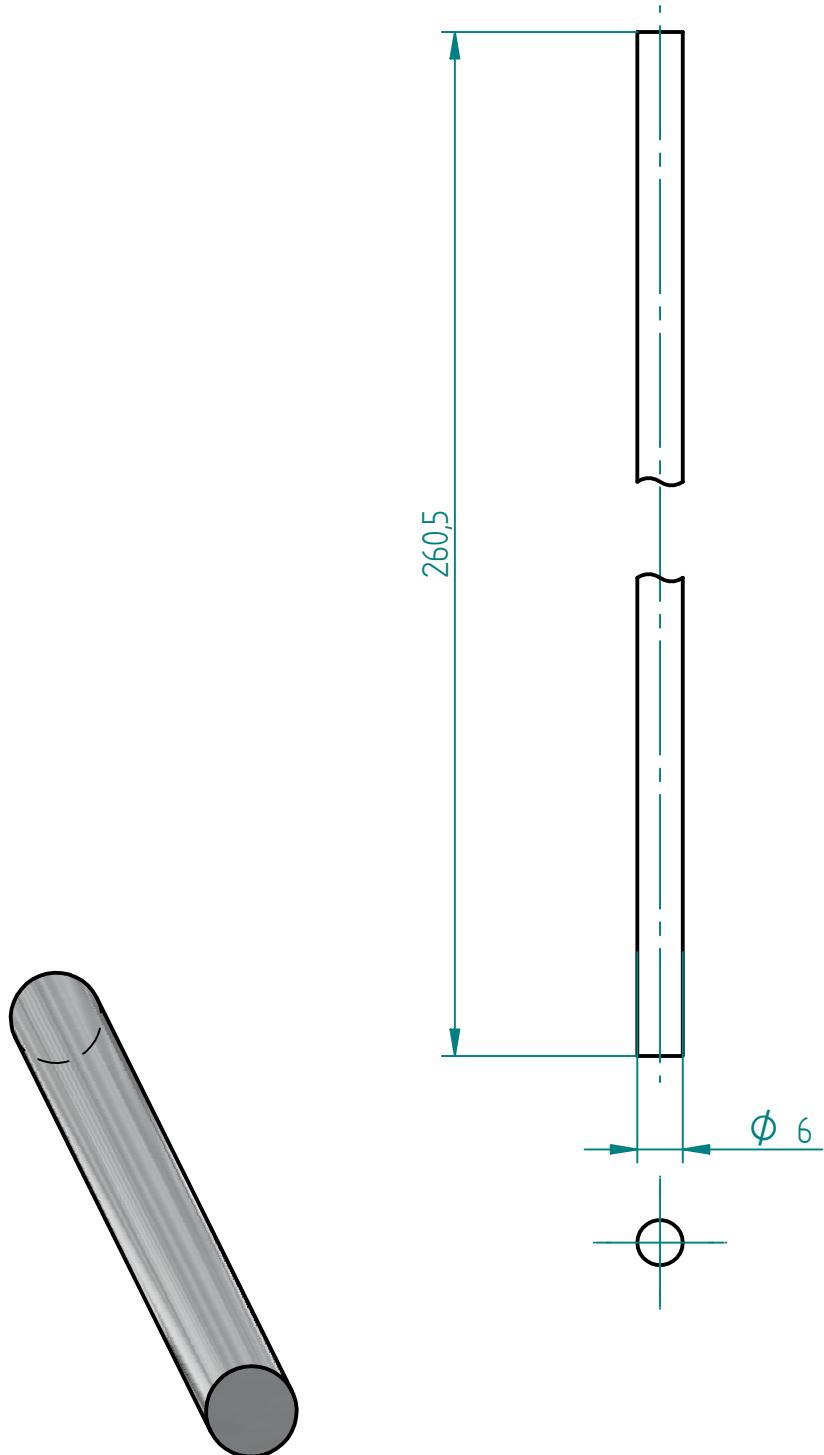


	NOMBRE	FECHA
DIBUJADO	M.I.G	22-02-15
MATERIAL	TRESPA	
ESPESOR	6 MM	
CANTIDAD	2 UNIDADES	
DIMENSIONES ACOTADAS EN MILIMETROS		
TOLERANCIA GENERAL $\pm 0.2$		
ANGULO $\pm 1^\circ$		

Proyecto Eurobot 2015

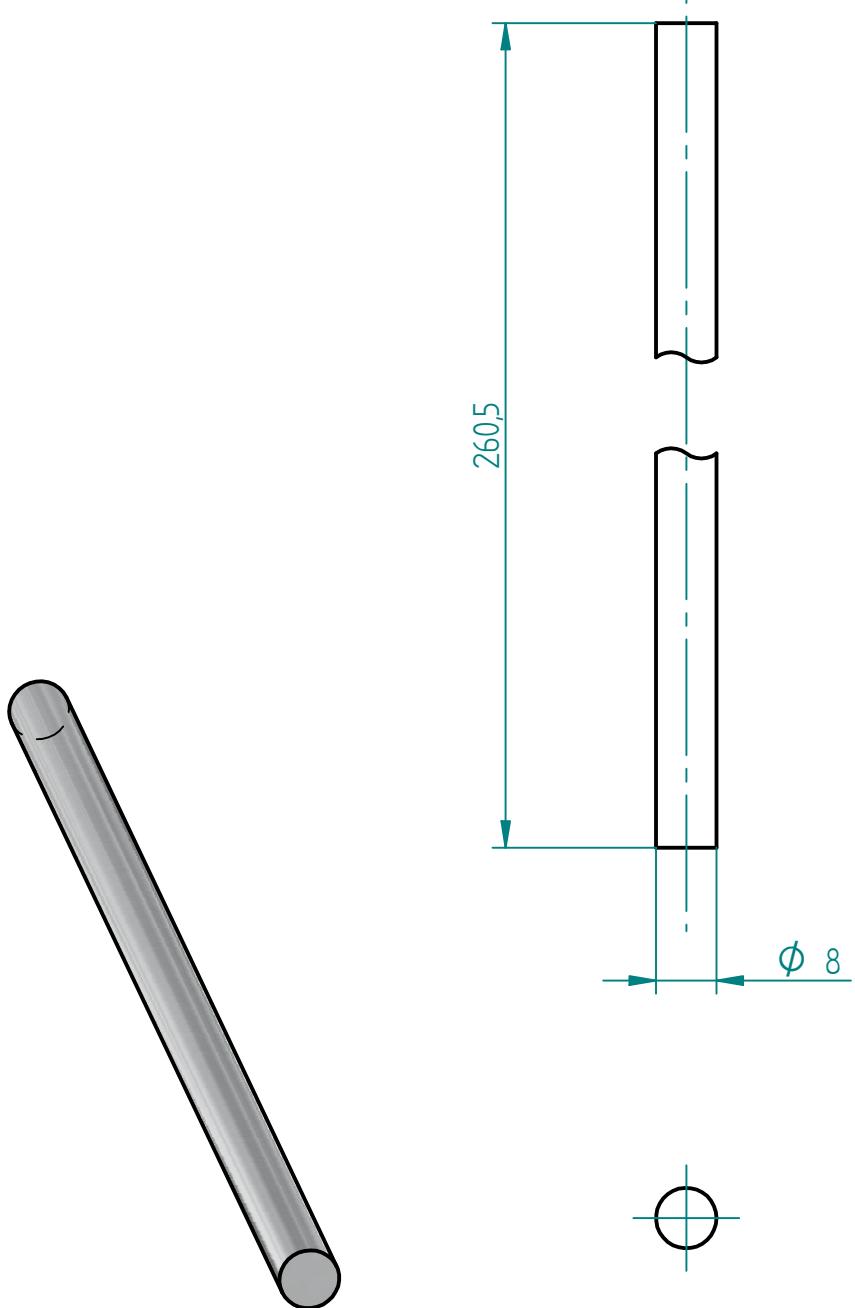
TITULO	HOJA		REV
TAPA BISAGRA SUPERIOR	A4	NUMERO PLANO 04-00-02-093	0
	ESCALA: 2:1	PESO:	HOJA 1 DE 1





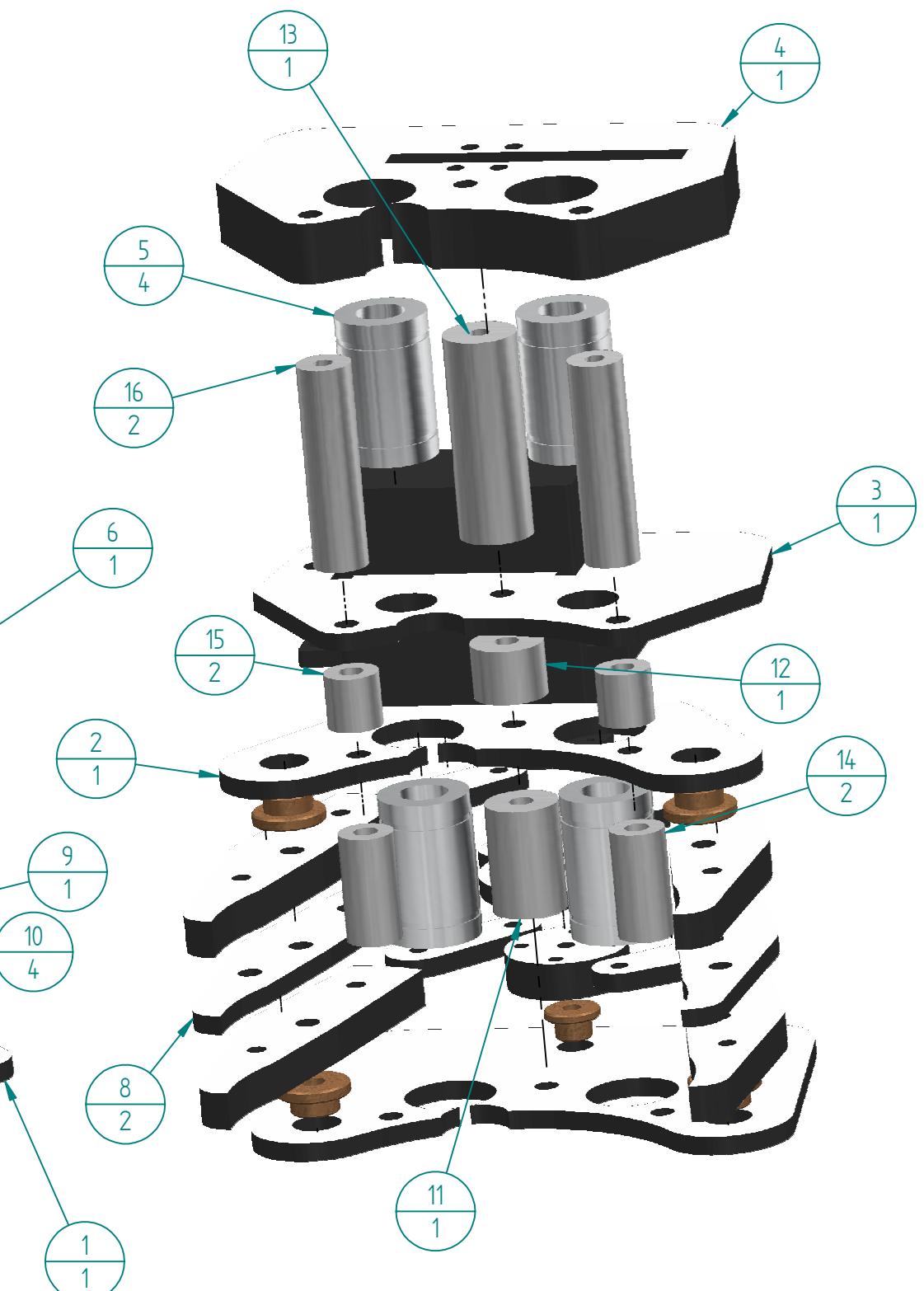
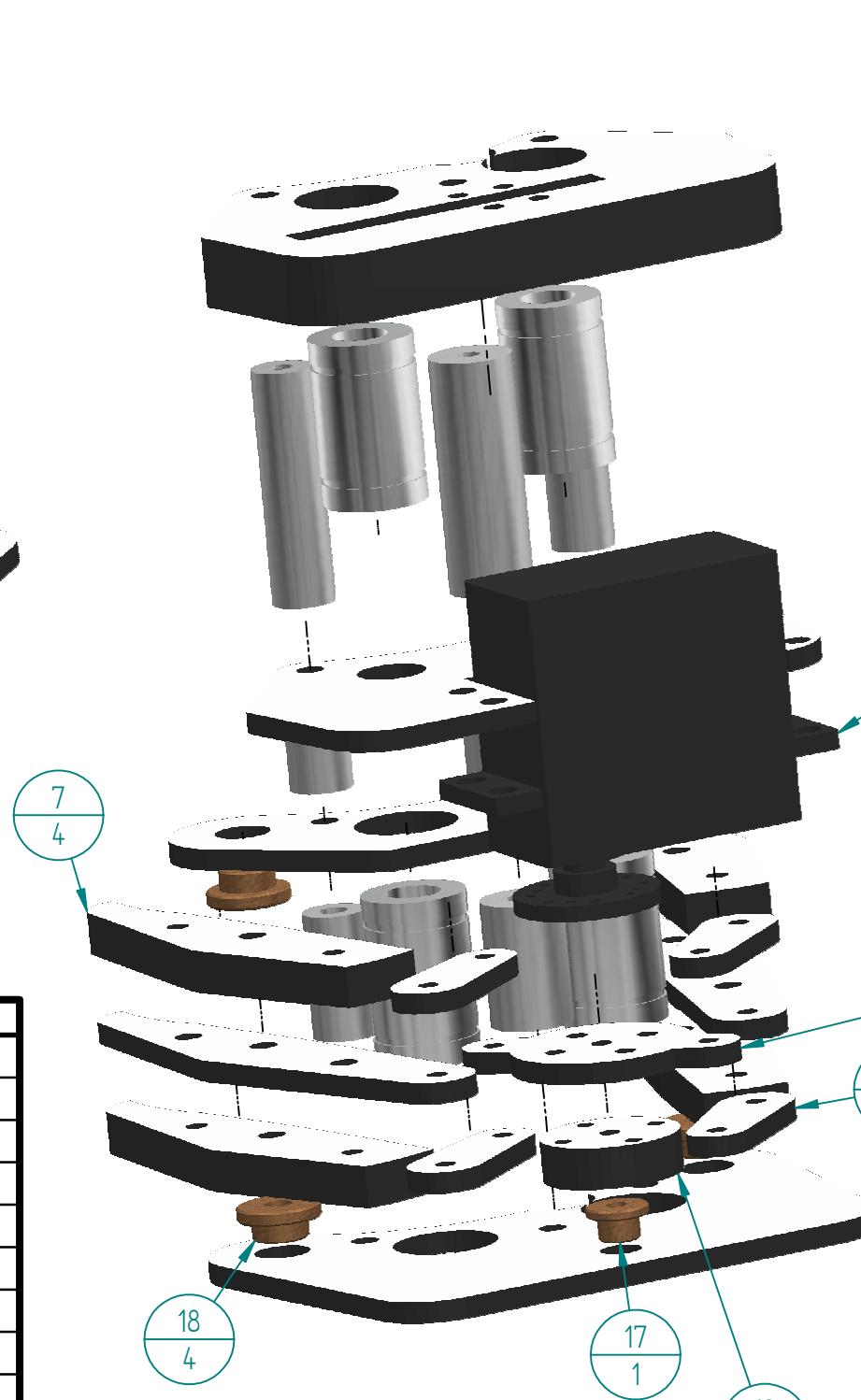
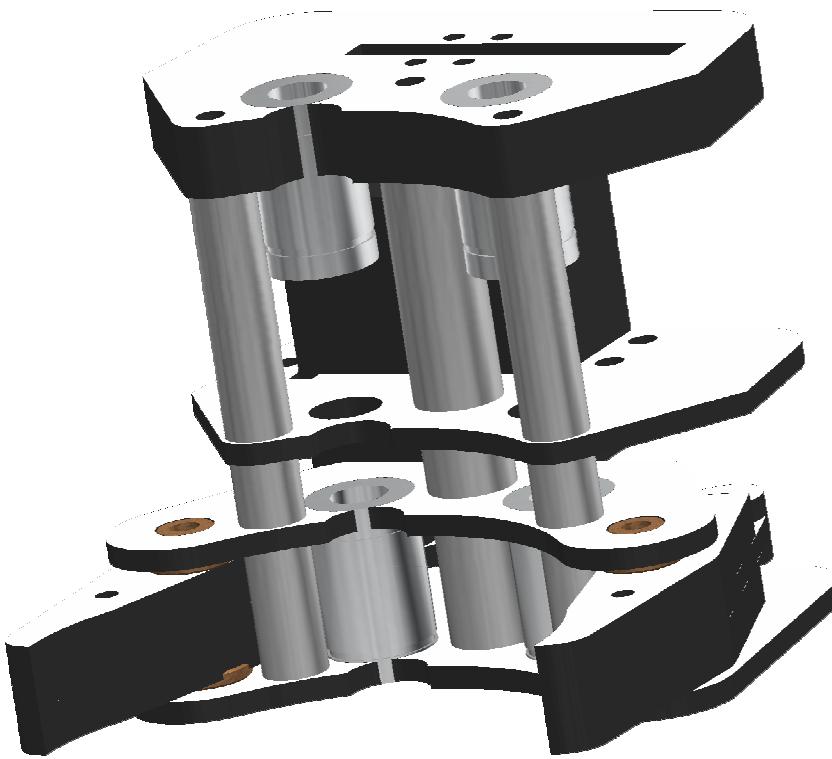
NOMBRE	M.I.G	FECHA	Proyecto Eurobot 2015		
DIBUJADO		XX-XX-XX			
MATERIAL	VARILLA ACERO PLATA			TITULO	
ESPESOR	$\Phi 6$ MM.			GUIAS 6 STAND	
CANTIDAD	4 UNIDADES				
DIMENSIONES ACOTADAS EN MILIMETROS			HOJA	NUMERO PLANO	REV
TOLERANCIA GENERAL $\pm 0.2$			A4	04-00-03-094	0
ANGULO $\pm 1^\circ$			ESCALA:	PESO:	HOJA 1 DE 1





NOMBRE		FECHA	Proyecto Eurobot 2015		
DIBUJADO	M.I.G	XX-XX-XX			
MATERIAL	VARILLA ACERO PLATA			TITULO	
ESPESOR	Ø 8 MM			GUIAS 8 CARROS	
CANTIDAD	4 UNIDADES				
DIMENSIONES ACOTADAS EN MILIMETROS			HOJA	NUMERO PLANO	REV
TOLERANCIA GENERAL ±0.2			A4	04-00-04-095	0
ANGULO ±1°			ESCALA:	PESO:	HOJA 1 DE 1

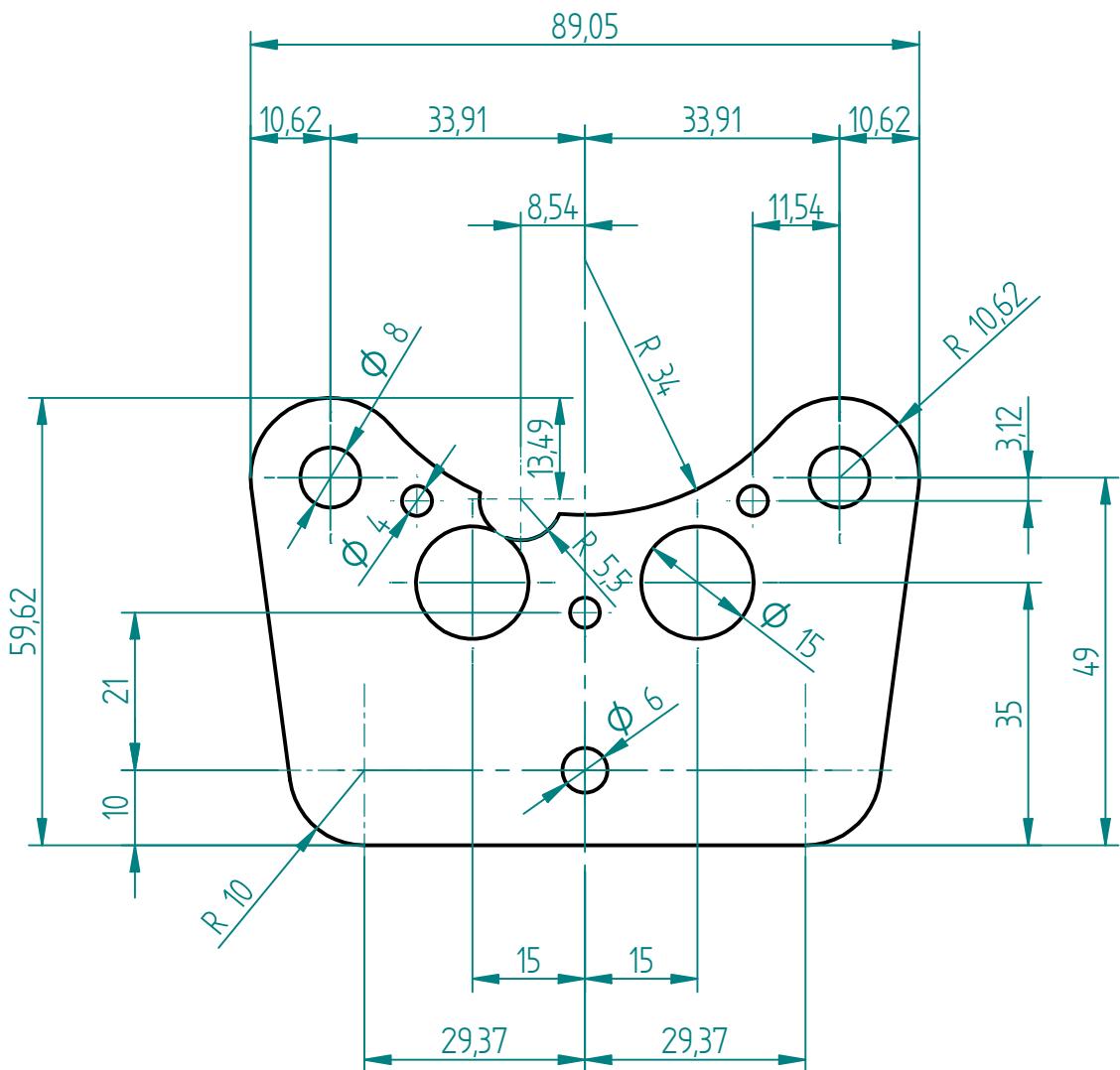




Item	PLANO	N	Material	Masa
1	Base 1 Carro Pinza	1	Epoxy, cast rigid	0,015 kg
2	Base 2 Carro Pinza	1	Epoxy, cast rigid	0,007 kg
3	Base 3 Carro Pinza	1	Epoxy, cast rigid	0,010 kg
4	Base 4 Carro Pinza	1	Epoxy, cast rigid	0,037 kg
5	Rodamiento lineal	4	Stainless steel	0,023 kg
6	Futaba 3003	1	(None)	0,037 kg
7	Pinza Suplemento	4	Epoxy, cast rigid	0,006 kg
8	Pinza Palanca	2	Epoxy, cast rigid	0,004 kg
9	Cabeza Servo Carro Pinza	1	Epoxy, cast rigid	0,002 kg
10	Pinza Leva	4	Epoxy, cast rigid	0,001 kg
11	Separador Central Base 1	1	Aluminum, 5050	0,005 kg
12	Separador Central Base 2	1	Aluminum, 5050	0,002 kg
13	Separador Central Base 3	1	Aluminum, 5050	0,010 kg
14	Separador Lateral Base 1	2	Aluminum, 5050	0,002 kg
15	Separador Lateral Base 2	2	Aluminum, 5050	0,001 kg
16	Separador Lateral Base 3	2	Aluminum, 5050	0,005 kg
17	Casquillo Autolubricado	1	Bronze, 90%	0,001 kg
18	Casquillo Autolubricado	4	Bronze, 90%	0,003 kg
19	Alojamiento Eje Servo	1	Bronze, 90%	0,015 kg

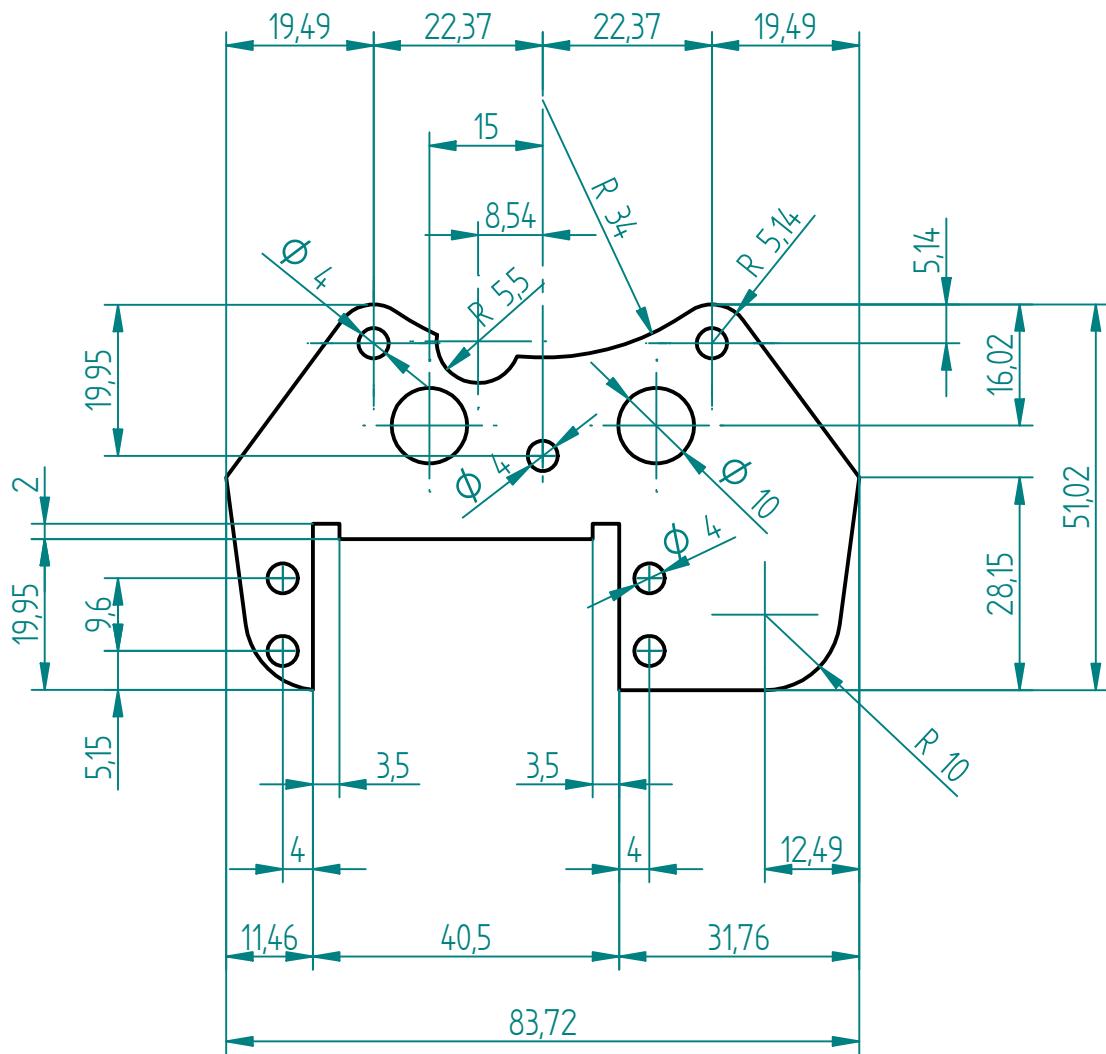
	NOMBRE	FECHA	Proyecto Eurobot 2015	
DIBUJADO	M.I.G	26-02-15		
MATERIAL	.			
ESPESOR	.			
CANTIDAD	1 S/D + 1 M/C			
DIMENSIONES ACOTADAS EN MILIMETROS		HOJA NUMERO PLANO		REV
TOLERANCIA GENERAL ±0.2		A3 04-01-00-E01		0
ANGULO ±1°		ESCALA:	PESO:	HOJA 1 DE 1





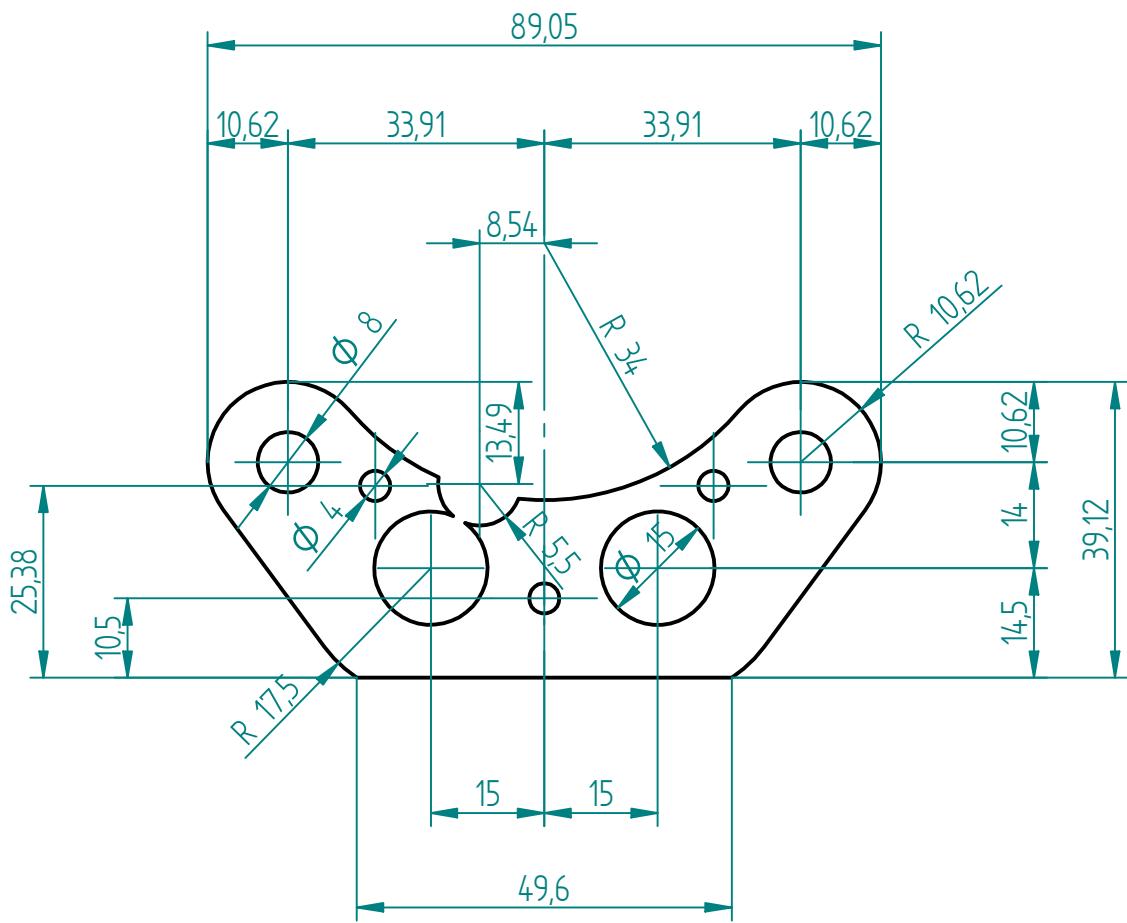
NOMBRE	M.I.G	FECHA	Proyecto Eurobot 2015		
DIBUJADO		18-01-15			
MATERIAL	TRESPA		TITULO		
ESPESOR	3 MM		BASE 1 CARRO PINZA		
CANTIDAD	2 UNIDADES				
DIMENSIONES ACOTADAS EN MILIMETROS TOLERANCIA GENERAL ±0.2 ANGULO ±1°			HOJA	NUMERO PLANO	REV
			A4	04-01-01-106	0
			ESCALA: 1:1	PESO:	HOJA 1 DE 1





NOMBRE	M.I.G	FECHA	Proyecto Eurobot 2015			
DIBUJADO		18-01-15				
MATERIAL	TRESPA		TITULO			
ESPESOR	3 MM		BASE 3 CARRO PINZA			
CANTIDAD	2 UNIDADES					
DIMENSIONES ACOTADAS EN MILIMETROS TOLERANCIA GENERAL ±0.2 ANGULO ±1°			HOJA	NUMERO PLANO	REV	
			A4	04-01-02-107	0	
			ESCALA:	PESO:	HOJA 1 DE 1	





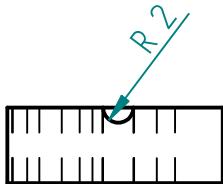
	NOMBRE	FECHA
DIBUJADO	M.I.G	18-01-15
MATERIAL	TRESPA	
ESPESOR	3 MM	
CANTIDAD	2 UNIDADES	
DIMENSIONES ACOTADAS EN MILIMETROS		
TOLERANCIA GENERAL $\pm 0.2$		
ANGULO $\pm 1^\circ$		

Proyecto Eurobot 2015

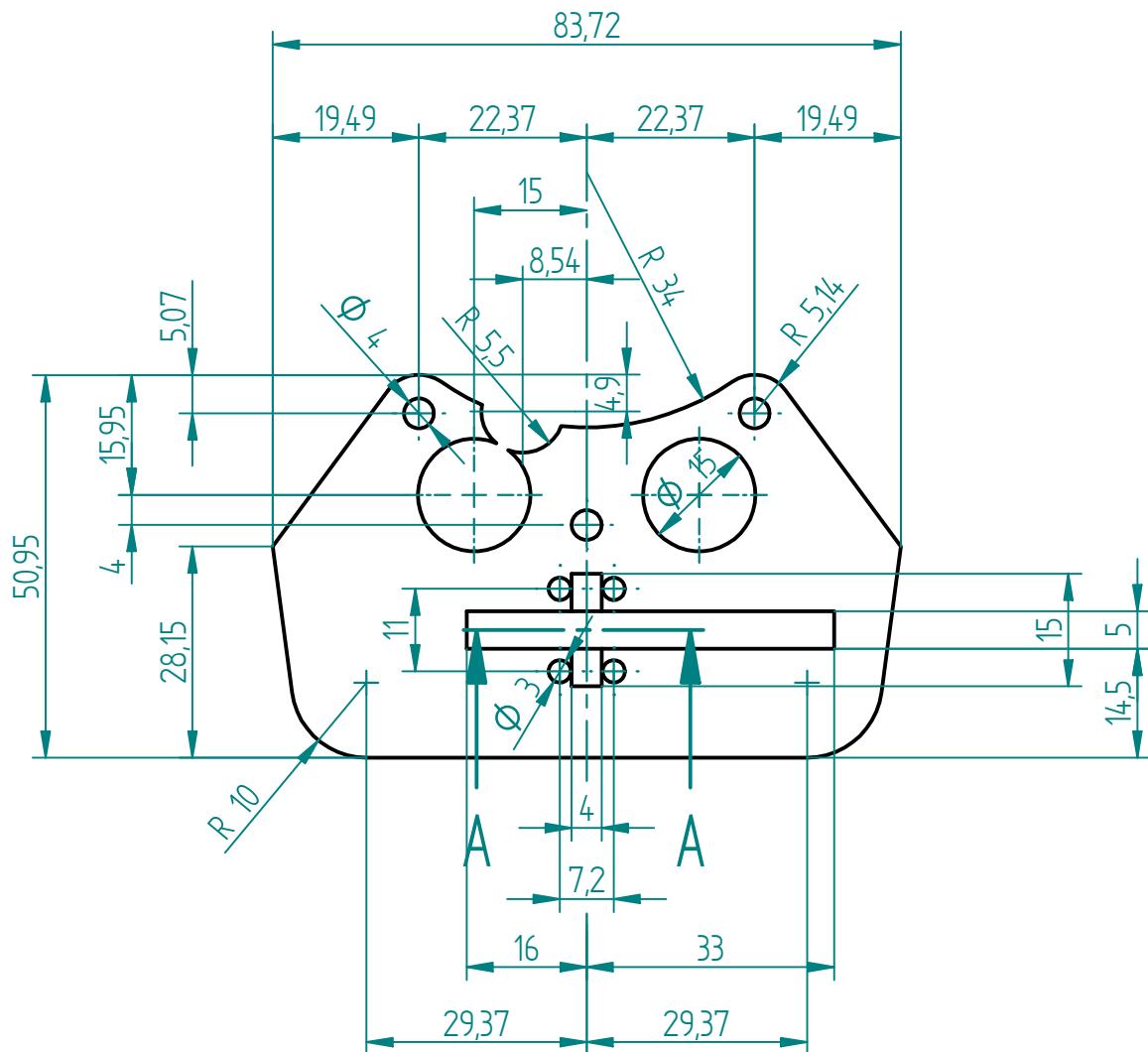
TITULO  
BASE 2 CARRO PINZA

HOJA A4	NUMERO PLANO 04-01-03-108	REV 0
ESCALA: 1:1	PESO:	HOJA 1 DE 1





## SECTION A-A



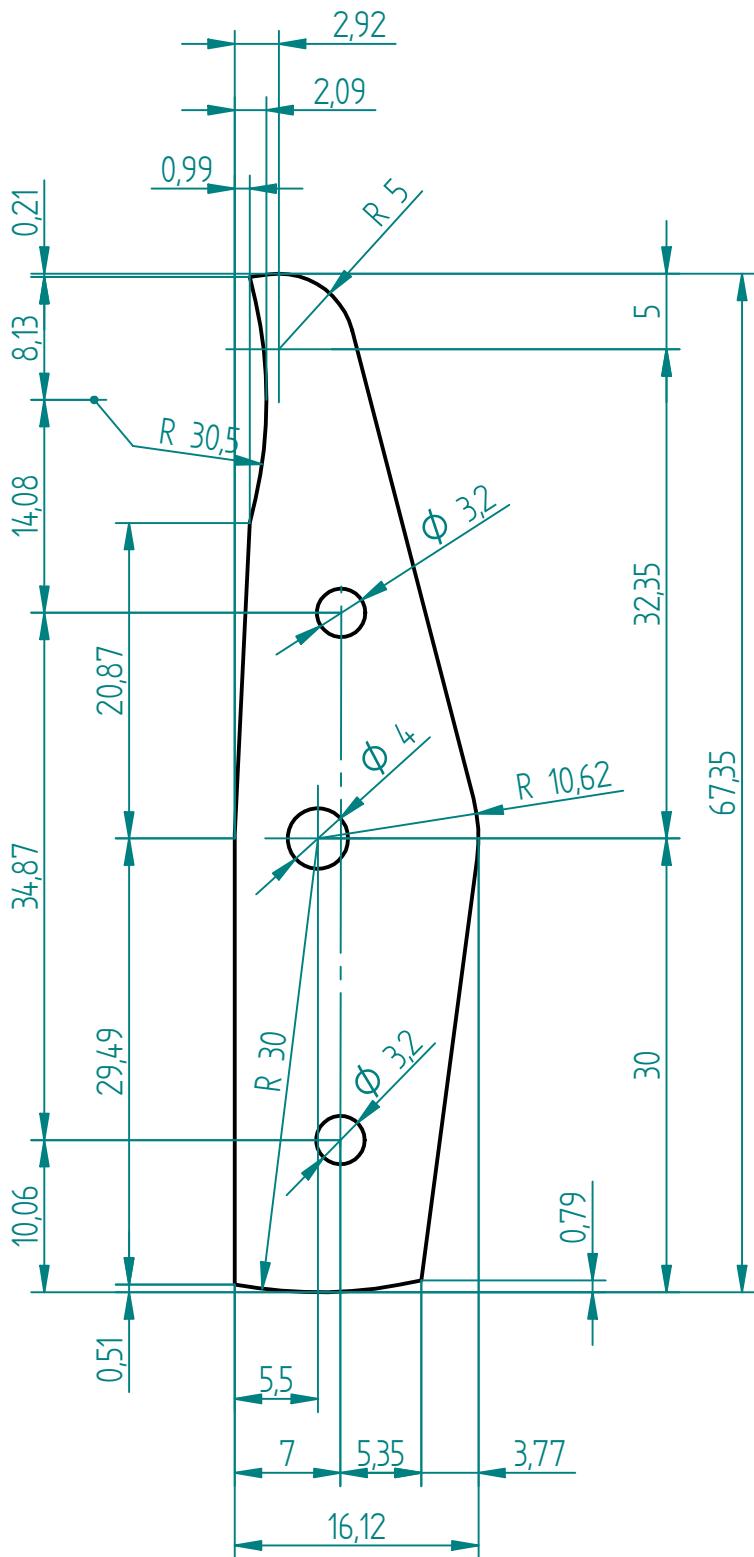
	NOMBRE	FECHA
DIBUJADO	M.I.G	18-01-15
MATERIAL	TRESPA	
ESPESOR	10 MM	
CANTIDAD	1 S/D + 1 M/C	
DIMENSIONES ACOTADAS EN MILIMETROS		
TOLERANCIA GENERAL $\pm 0.2$		
ANGULO $\pm 1^\circ$		

Proyecto Eurobot 2015

TITULO  
BASE 4 CARRO PINZA

HOJA	NUMERO PLANO	REV
A4	04-01-04-109	0
ESCALA: 1:1	PESO:	HOJA 1 DE 1





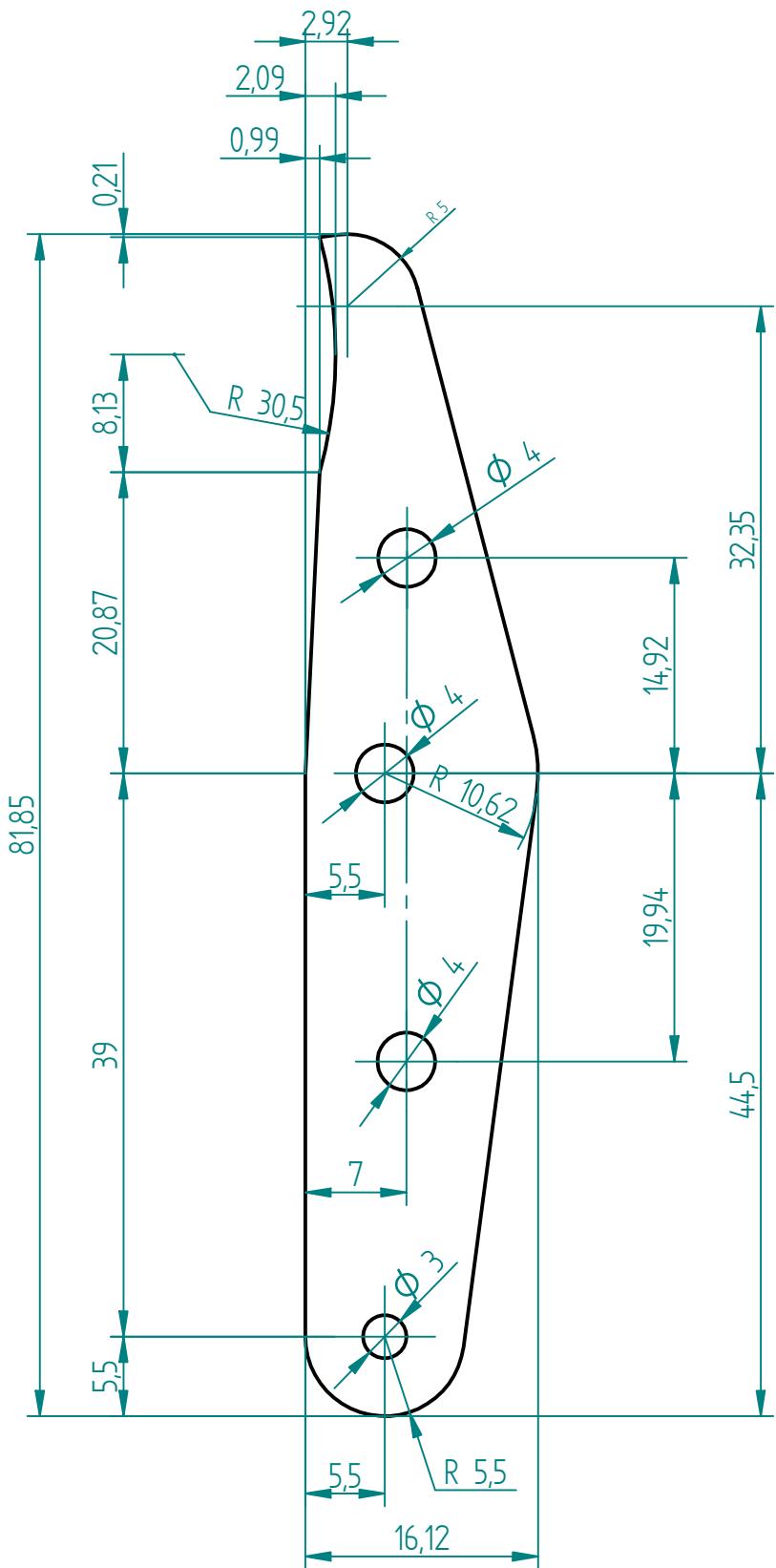
	NOMBRE	FECHA
DIBUJADO	M.I.G	18-01-15
MATERIAL	TRESPA	
ESPESOR	6 MM	
CANTIDAD	8 UNIDADES	
DIMENSIONES ACOTADAS EN MILIMETROS		
TOLERANCIA GENERAL ±0.2		
ANGULO ±1°		

Proyecto Eurobot 2015

TITULO PINZA SUPLEMENTO

HOJA A4	NUMERO PLANO 04-01-05-110	REV 0
ESCALA: 1:1	PESO:	HOJA 1 DE 1



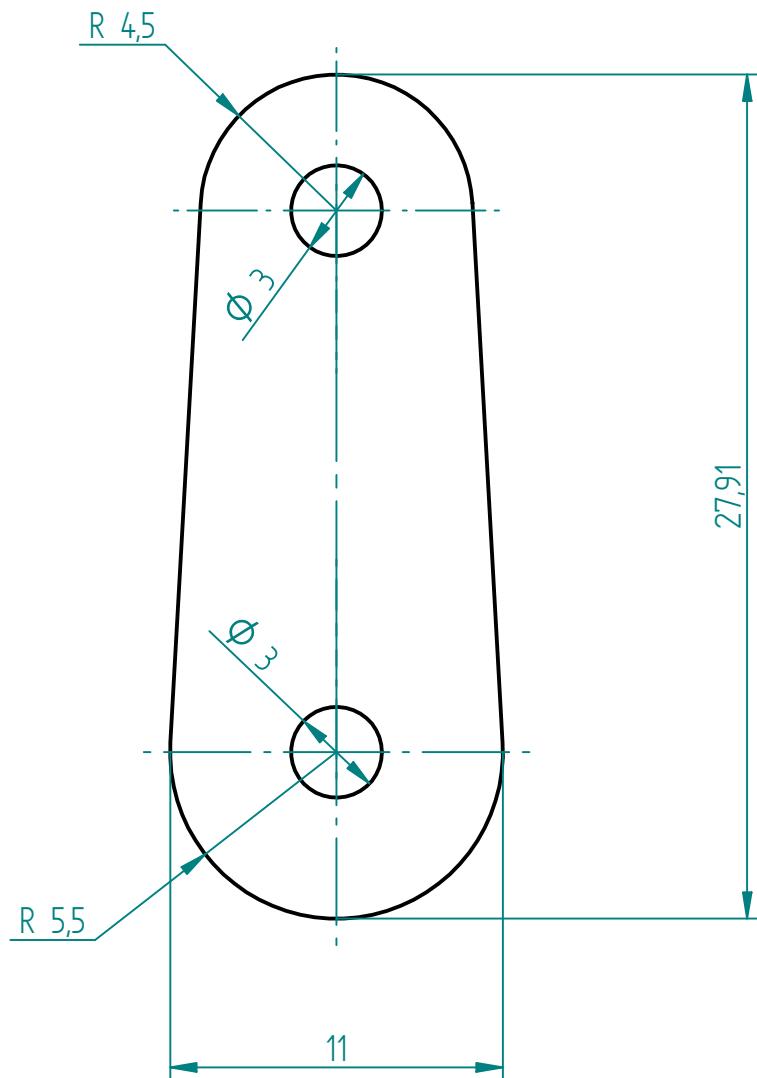


	NOMBRE	FECHA
DIBUJADO	M.I.G	18-01-15
MATERIAL	TRESPA	
ESPESOR	3 MM	
CANTIDAD	4 UNIDADES	
DIMENSIONES ACOTADAS EN MILIMETROS		
TOLERANCIA GENERAL $\pm 0.2$		
ANGULO $\pm 1^\circ$		

Proyecto Eurobot 2015

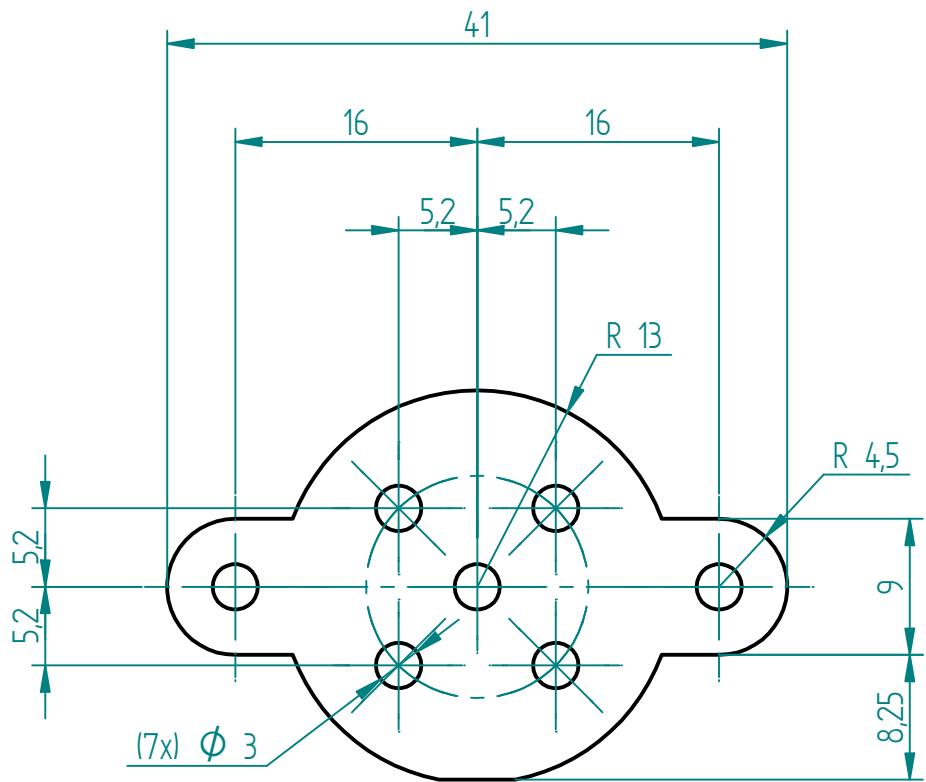
TITULO	HOJA	NUMERO PLANO	REV
PINZA PALANCA PAR	A4	04-01-06-111	0
	ESCALA: 1:1	PESO:	HOJA 1 DE 1





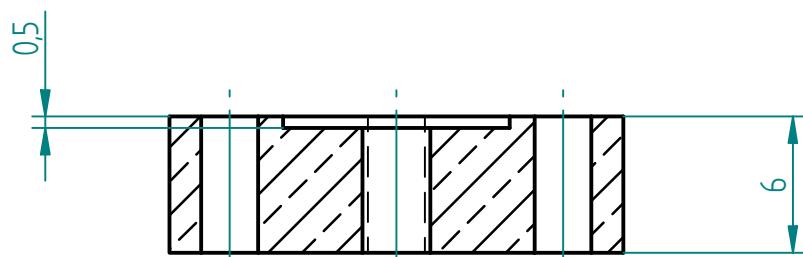
NOMBRE	FECHA	Proyecto Eurobot 2015		
DIBUJADO	M.I.G	18-01-15		
MATERIAL	TRESPA		TITULO	
ESPESOR	3 MM		PINZA LEVA	
CANTIDAD	8 UNIDADES			
DIMENSIONES ACOTADAS EN MILIMETROS		HOJA	NUMERO PLANO	REV
TOLERANCIA GENERAL ±0.2		A4	04-01-07-112	0
ANGULO ±1°		ESCALA: 4:1	PESO:	HOJA 1 DE 1



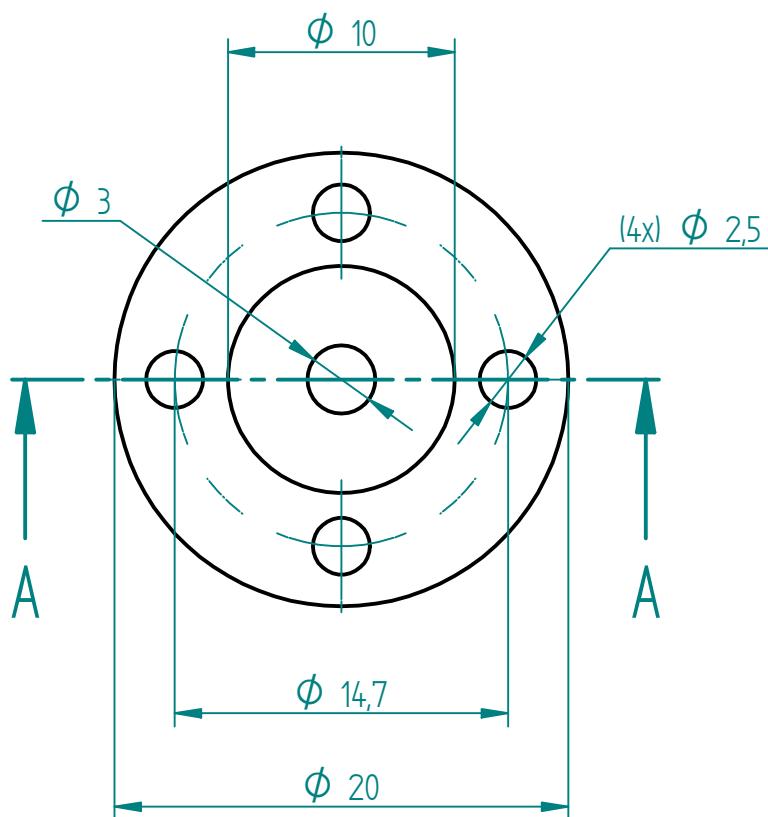


NOMBRE	M.I.G	FECHA	Proyecto Eurobot 2015		
DIBUJADO		18-01-15			
MATERIAL	TRESPA		TITULO		
ESPESOR	3 MM		CABEZA SERVO CARRO PINZA		
CANTIDAD	2 UNIDADES				
DIMENSIONES ACOTADAS EN MILIMETROS			HOJA	NUMERO PLANO	REV
TOLERANCIA GENERAL ±0.2			A4	04-01-08-113	0
ANGULO ±1°			ESCALA:	PESO:	HOJA 1 DE 1



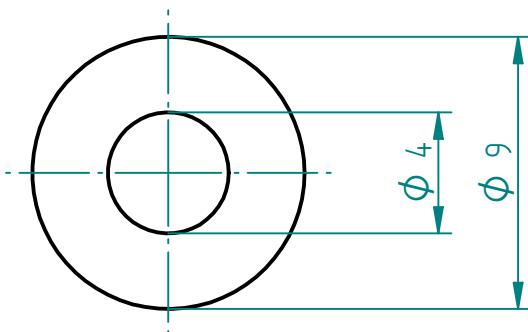
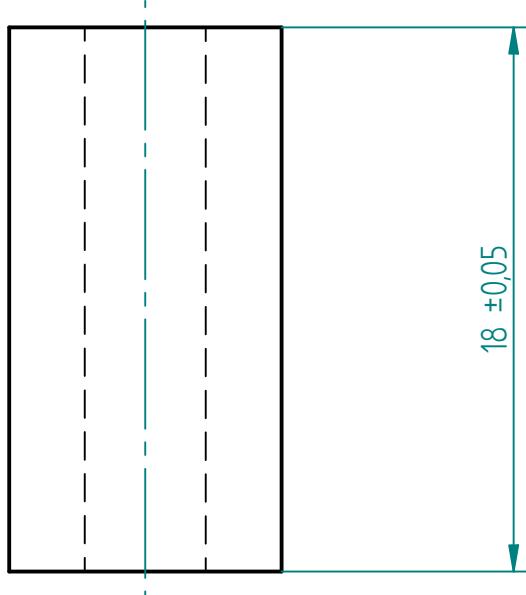


SECTION A-A



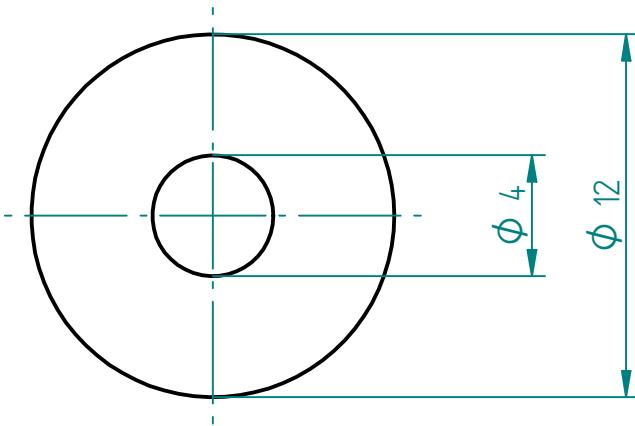
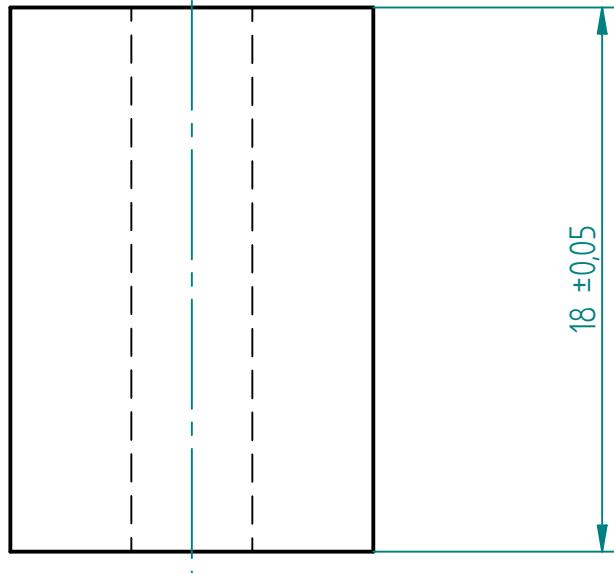
NOMBRE	M.I.G	FECHA	Proyecto Eurobot 2015		
DIBUJADO		26-02-15			
MATERIAL	TRESPA		TITULO		
ESPESOR	6 MM		ALOJAMIENTO EJE SERVO		
CANTIDAD	2 UNIDADES				
DIMENSIONES ACOTADAS EN MILIMETROS TOLERANCIA GENERAL ±0.2 ANGULO ±1°			HOJA	NUMERO PLANO	REV
			A4	04-01-09-114	0
			ESCALA:	PESO:	HOJA 1 DE 1





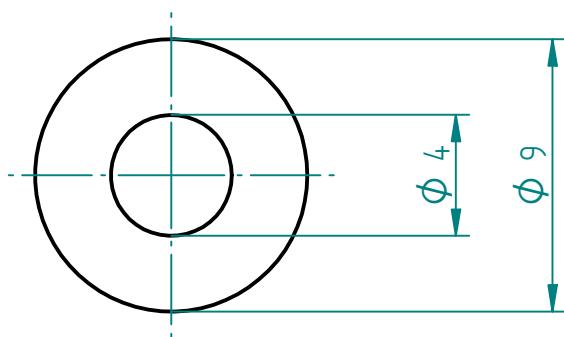
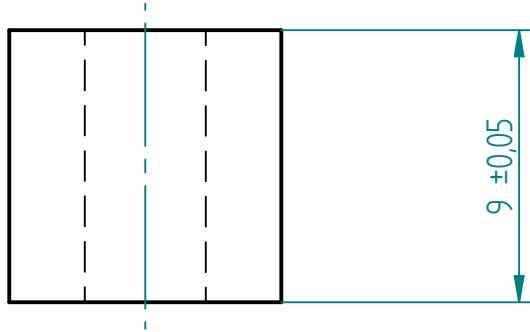
NOMBRE	FECHA	Proyecto Eurobot 2015		
DIBUJADO	M.I.G	18-01-15		
MATERIAL	ALUMINIO			
ESPESOR	Ø 9 MM			
CANTIDAD	4 UNIDADES			
DIMENSIONES ACOTADAS EN MILIMETROS TOLERANCIA GENERAL ±0.2 ANGULO ±1°		HOJA A4	NUMERO PLANO 04-01-10-115	REV 0
		ESCALA: 4:1	PESO:	HOJA 1 DE 1





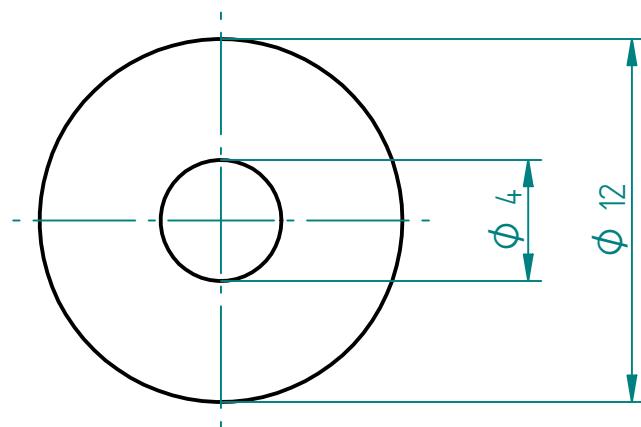
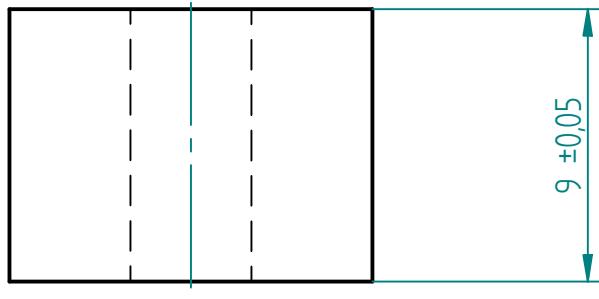
NOMBRE	FECHA	Proyecto Eurobot 2015		
DIBUJADO	18-01-15			
MATERIAL	ALUMINIO	TITULO		
ESPESOR	$\Phi 12$ MM	SOPORTE CENTRAL BASE 1		
CANTIDAD	2 UNIDADES	CARRO PINZA		
DIMENSIONES ACOTADAS EN MILIMETROS		HOJA	NUMERO PLANO	REV
TOLERANCIA GENERAL $\pm 0,2$		A4	04-01-11-116	0
ANGULO $\pm 1^\circ$		ESCALA:	4:1	PESO:
				HOJA 1 DE 1





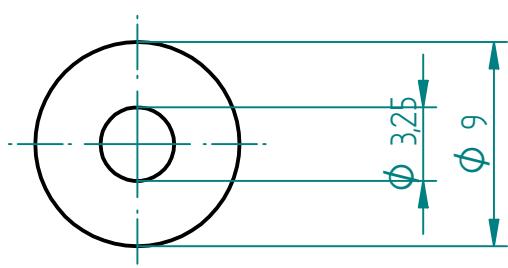
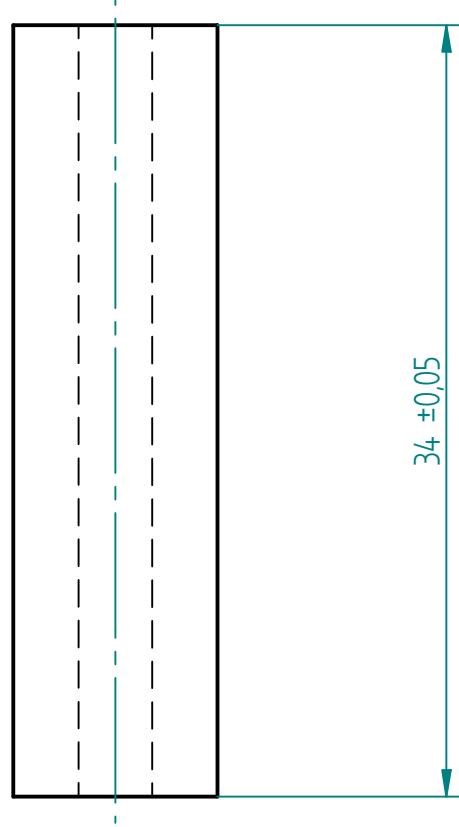
NOMBRE	FECHA	Proyecto Eurobot 2015		
DIBUJADO	M.I.G	18-01-15		
MATERIAL	ALUMINIO			
ESPESOR	Ø 9 MM			
CANTIDAD	4 UNIDADES			
DIMENSIONES ACOTADAS EN MILIMETROS		HOJA	NUMERO PLANO	REV
TOLERANCIA GENERAL ±0.2		A4	04-01-12-117	0
ANGULO ±1°		ESCALA:	4:1	PESO:
				HOJA 1 DE 1





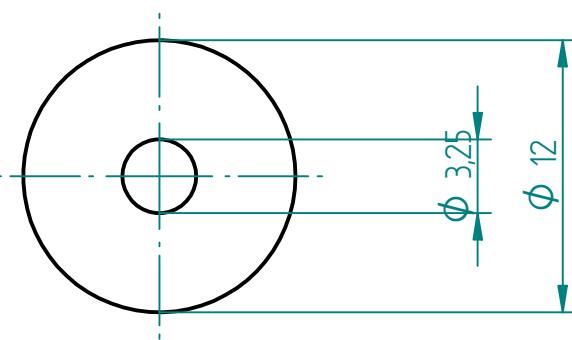
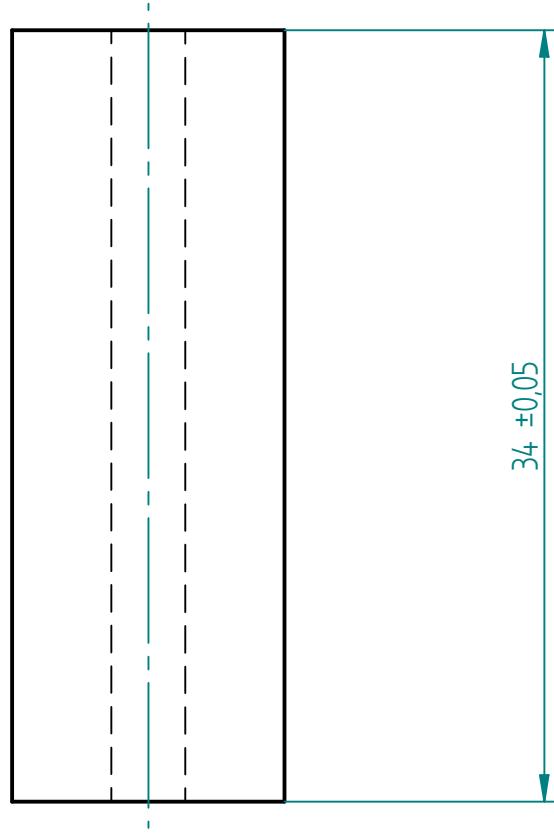
NOMBRE	FECHA	Proyecto Eurobot 2015		
DIBUJADO	M.I.G	18-01-15		
MATERIAL	ALUMINIO			
ESPESOR	Ø 12 MM			
CANTIDAD	2 UNIDADES			
DIMENSIONES ACOTADAS EN MILIMETROS TOLERANCIA GENERAL ±0.2 ANGULO ±1°		HOJA A4	NUMERO PLANO 04-01-13-118	REV 0
		ESCALA: 4:1	PESO:	HOJA 1 DE 1





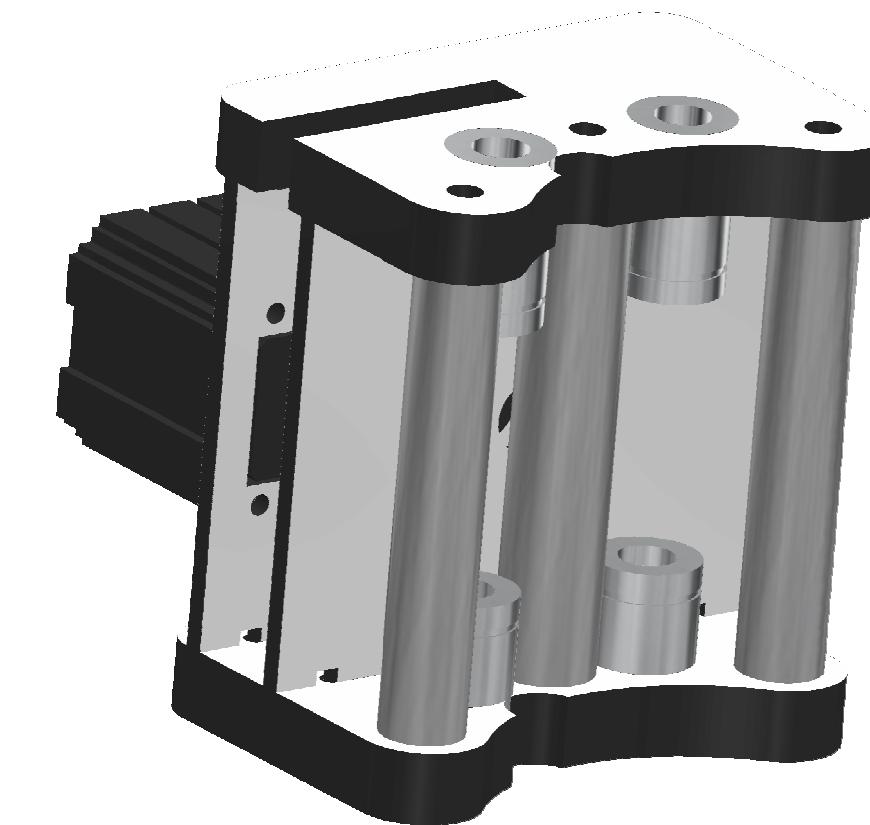
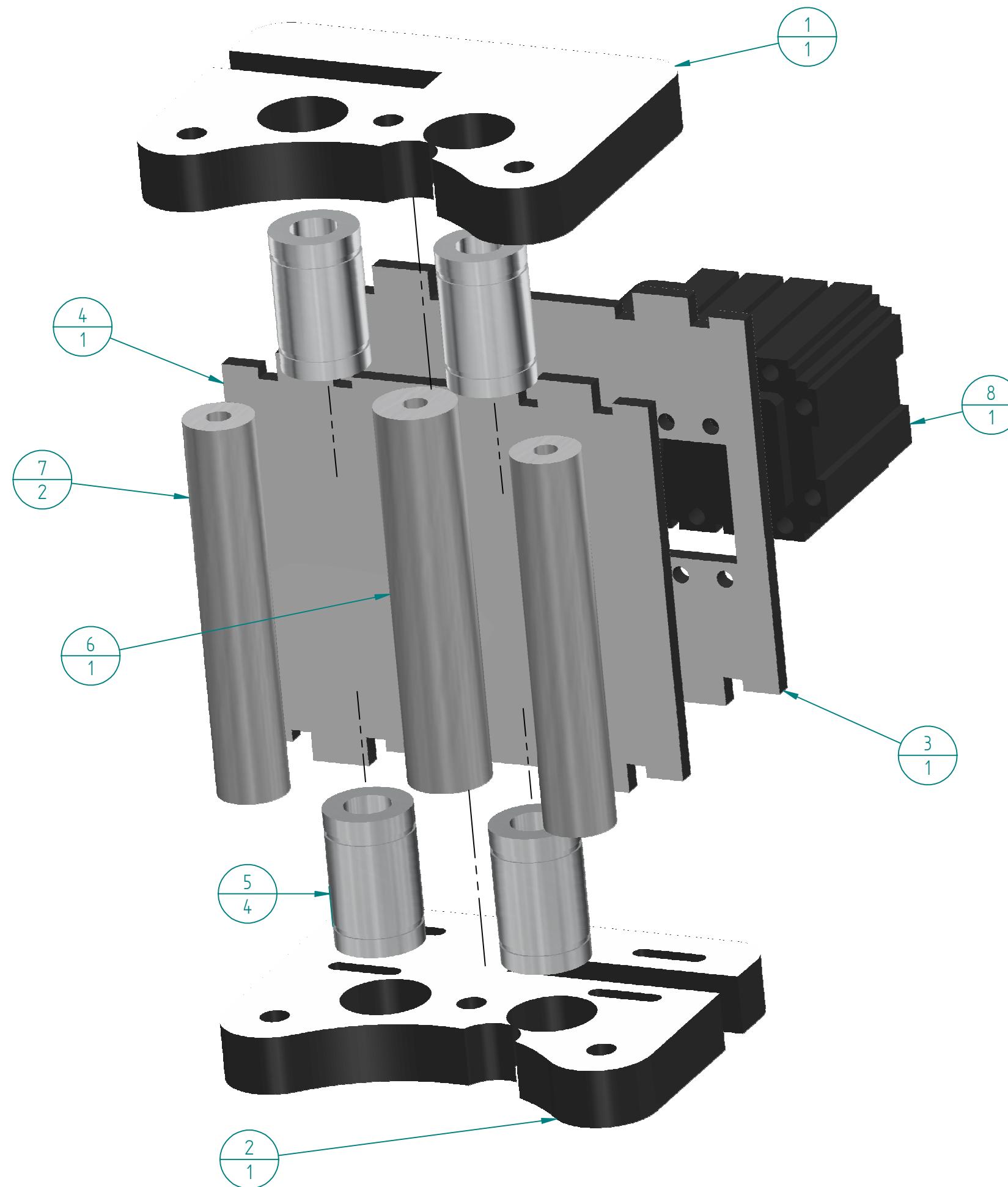
NOMBRE	FECHA	Proyecto Eurobot 2015		
DIBUJADO	M.I.G	18-01-15		
MATERIAL	ALUMINIO			
ESPESOR	$\phi 9$ MM			
CANTIDAD	4 UNIDADES			
DIMENSIONES ACOTADAS EN MILIMETROS		HOJA	NUMERO PLANO	
TOLERANCIA GENERAL $\pm 0.2$		A4	04-01-14-119	REV 0
ANGULO $\pm 1^\circ$		ESCALA:	3:1	PESO:
				HOJA 1 DE 1





NOMBRE	FECHA	Proyecto Eurobot 2015		
DIBUJADO	M.I.G	18-01-15		
MATERIAL	ALUMINIO			
ESPESOR	$\phi 12$ MM			
CANTIDAD	2 UNIDADES			
DIMENSIONES ACOTADAS EN MILIMETROS TOLERANCIA GENERAL $\pm 0.2$ ANGULO $\pm 1^\circ$		HOJA A4	NUMERO PLANO 04-01-15-120	REV 0
		ESCALA: 4:1	PESO:	HOJA 1 DE 1

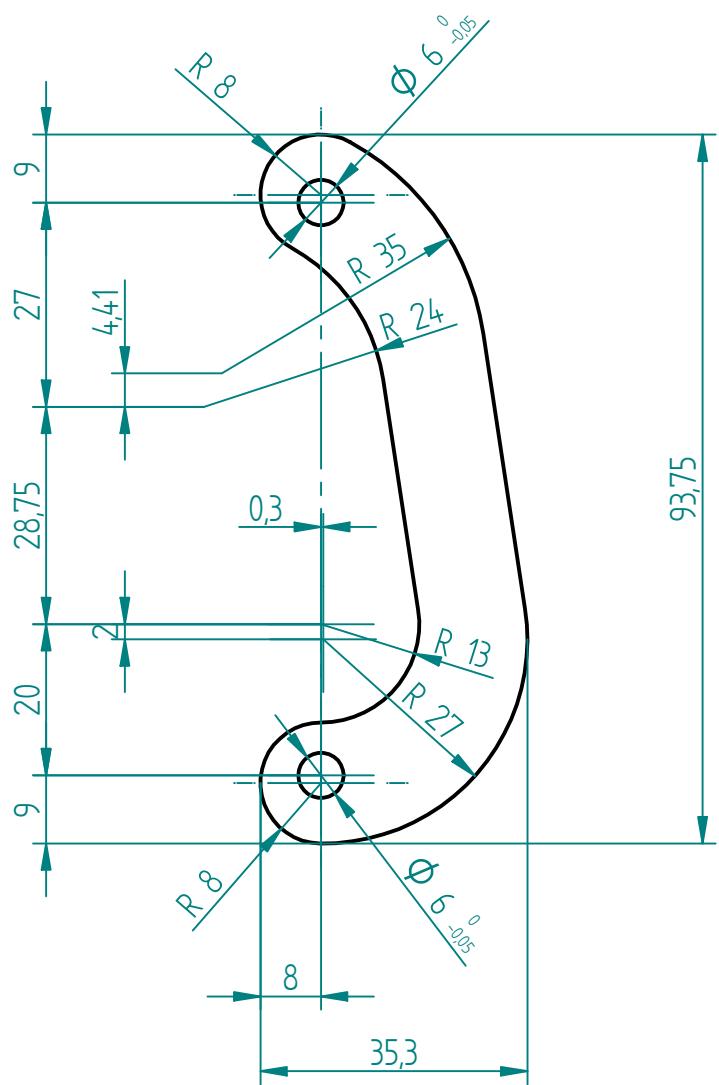




Item	PLANO	N	Material	MASA
1	Base Carro Servo Derecha	1	Epoxy, cast	0,039 kg
2	Tapa Carro Servo Derecha	1	Epoxy, cast	0,039 kg
3	Soporte Servo Carro Servo	1	Epoxy, cast	0,016 kg
4	Soporte Eje Servo Carro	1	Epoxy, cast	0,020 kg
5	Rodamiento Lineal	4	Stainless steel	0,023 kg
6	Separador Central Carro	1	Aluminum, 5050	0,025 kg
7	Separador Lateral Carro	2	Aluminum, 5050	0,018 kg
8	Dynamixel Simplificado	1	(None)	0,054 kg

NOMBRE	FECHA	Proyecto Eurobot 2015	
DIBUJADO	M.I.G	18-01-15	
MATERIAL	.		
ESPESOR	.		
CANTIDAD	1 S/D + 1 M/C	TITULO CARRO SERVO 2 (EXPLOSION)	
DIMENSIONES ACOTADAS EN MILIMETROS TOLERANCIA GENERAL ±0.2 ANGULO ±1°		HOJA	NUMERO PLANO
		A3	04-02-00-E01
		REV 0	
ESCALA:		PESO:	HOJA 1 DE 1





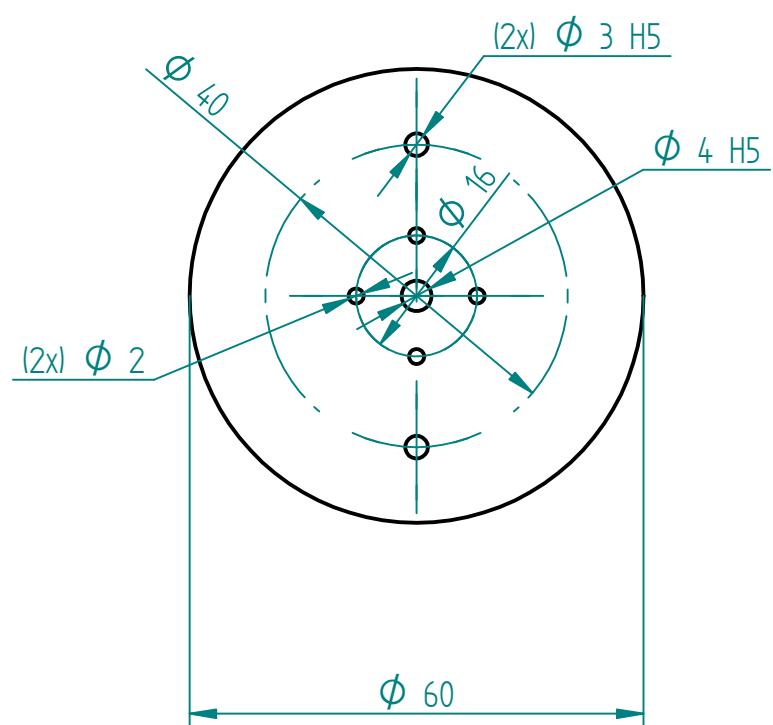
	NOMBRE	FECHA
DIBUJADO	M.I.G	18-01-15
MATERIAL	TRESPA	
ESPESOR	3 MM	
CANTIDAD	4 UNIDADES	
DIMENSIONES ACOTADAS EN MILIMETROS		
TOLERANCIA GENERAL $\pm 0.2$		
ANGULO $\pm 1^\circ$		

Proyecto Eurobot 2015

TITULO  
PALANCA CARRO SERVO

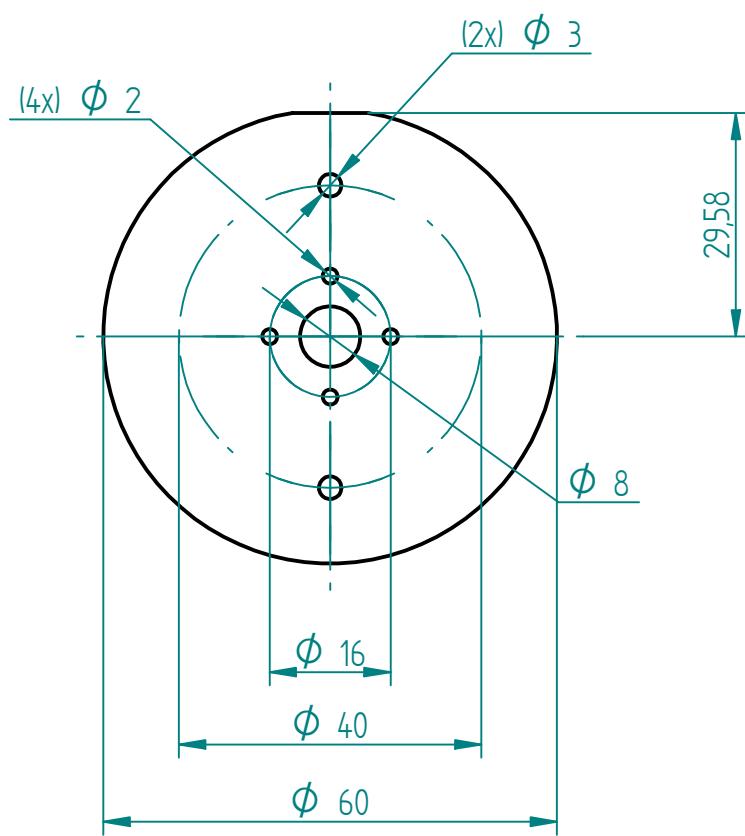
HOJA A4	NUMERO PLANO 04-02-01-096	REV 0
ESCALA: 1:1	PESO:	HOJA 1 DE 1





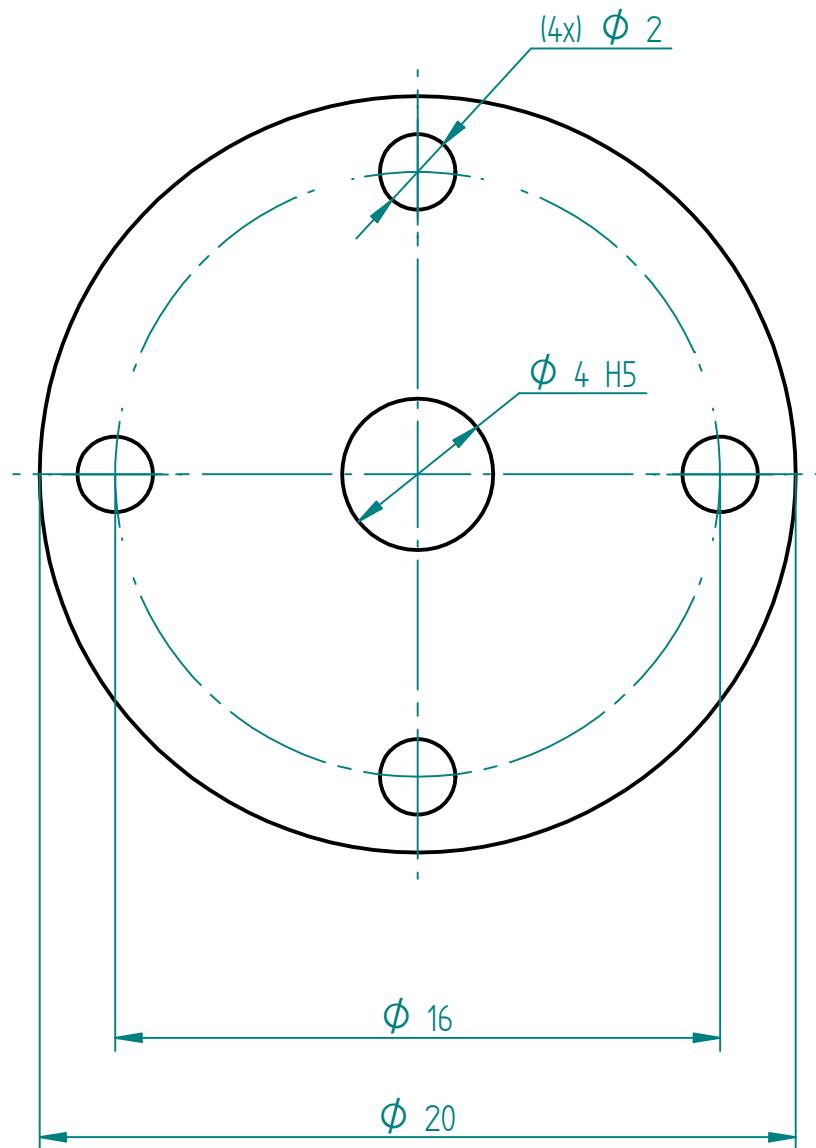
NOMBRE		FECHA	Proyecto Eurobot 2015		
DIBUJADO	M.I.G	18-01-15			
MATERIAL	TRESPA				TITULO
ESPESOR	3 MM				PLATO CARRO SERVO LADO SOPORTE EJE
CANTIDAD	2 UNIDADES				
DIMENSIONES ACOTADAS EN MILIMETROS			HOJA	NUMERO PLANO	REV
TOLERANCIA GENERAL ±0.2			A4	04-02-02-097	0
ANGULO ±1°			ESCALA:	1:1	PESO:
					HOJA 1 DE 1





NOMBRE	FECHA	Proyecto Eurobot 2015		
DIBUJADO	M.I.G	18-01-15		
MATERIAL	TRESPA	TITULO		
ESPESOR	3 MM	PLATO CARRO SERVO LADO SERVO		
CANTIDAD	2 UNIDADES			
DIMENSIONES ACOTADAS EN MILIMETROS		HOJA	NUMERO PLANO	REV
TOLERANCIA GENERAL $\pm 0.2$		A4	04-02-03-098	0
ANGULO $\pm 1^\circ$		ESCALA: 1:1	PESO:	HOJA 1 DE 1

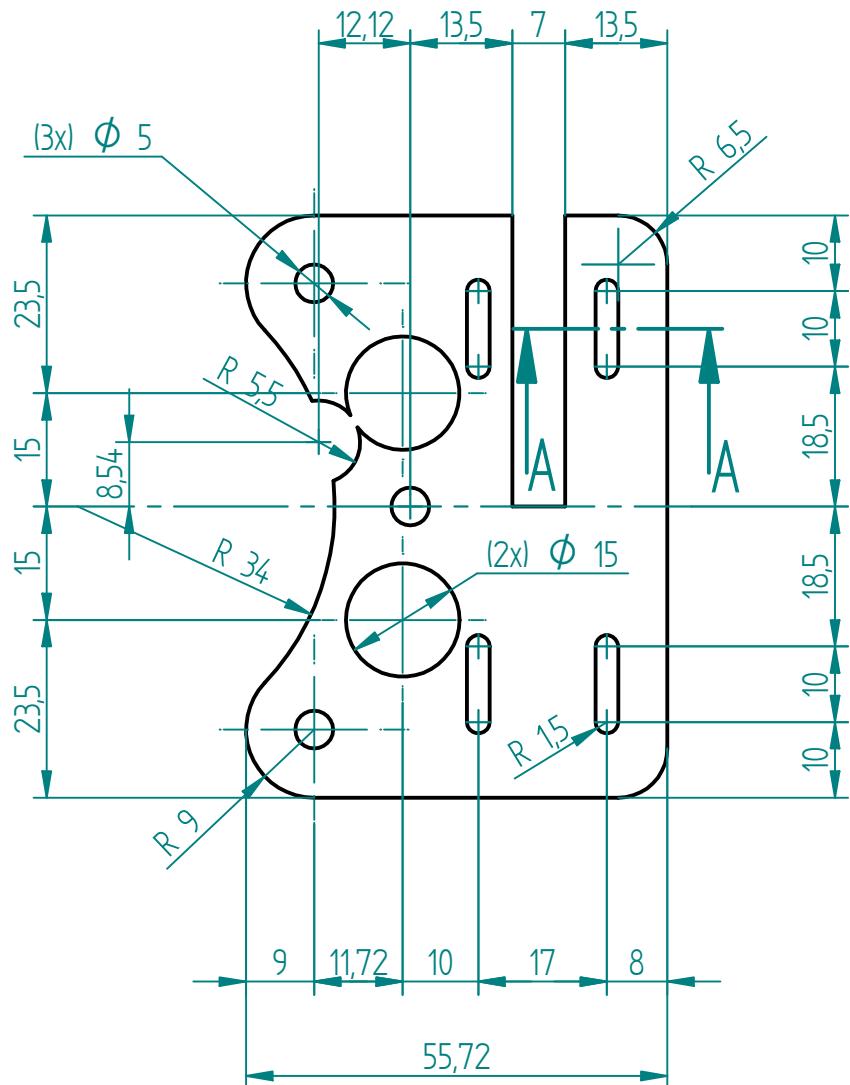
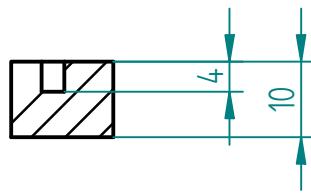




NOMBRE	FECHA	Proyecto Eurobot 2015		
DIBUJADO	M.I.G	18-01-15		
MATERIAL	TRESPA	TITULO		
ESPESOR	5 MM	SEPARADOR PLATOS CARRO SERVO		
CANTIDAD	2 UNIDADES			
DIMENSIONES ACOTADAS EN MILIMETROS		HOJA	NUMERO PLANO	REV
TOLERANCIA GENERAL $\pm 0.2$		A4	04-02-04-099	0
ANGULO $\pm 1^\circ$		ESCALA: 5:1	PESO:	HOJA 1 DE 1



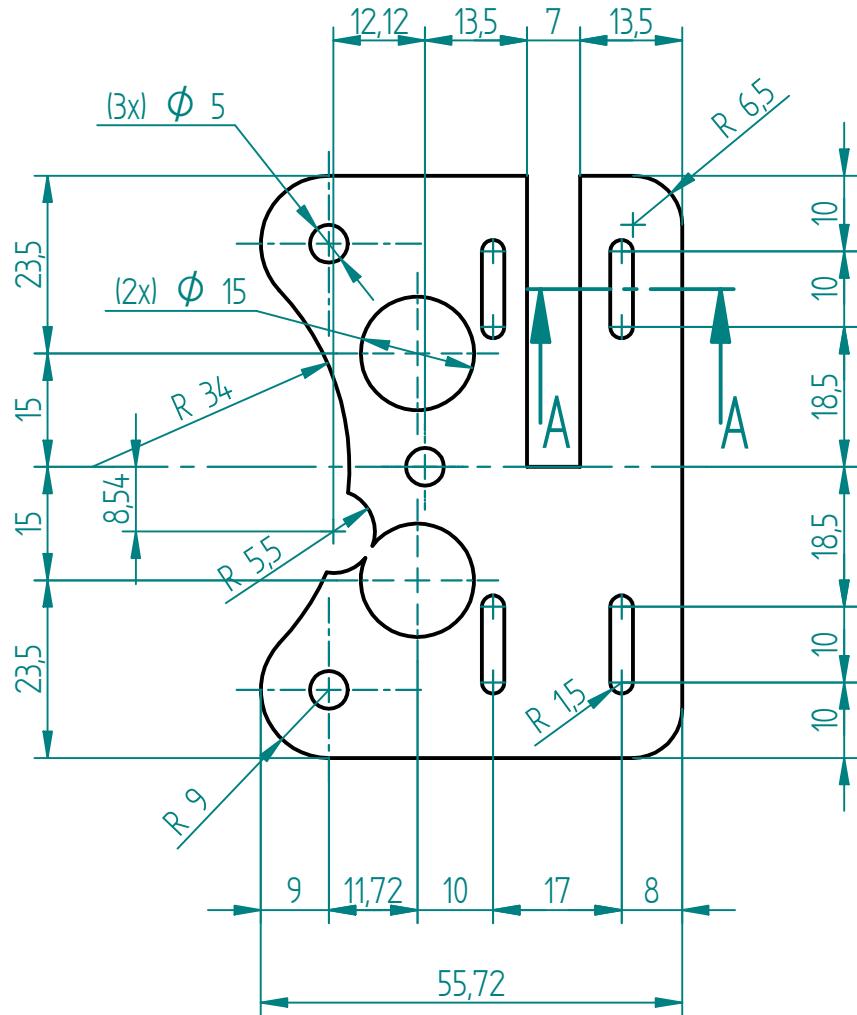
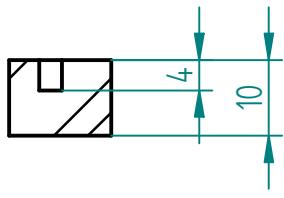
## SECTION A-A



NOMBRE	M.I.G	FECHA	Proyecto Eurobot 2015		
DIBUJADO		28-02-15			
MATERIAL	TRESPA				
ESPESOR	10 MM				
CANTIDAD	1 S/D + 1 M/C	TITULO			
DIMENSIONES ACOTADAS EN MILIMETROS			TAPA CARRO SERVO DERECHA MIR 1		
TOLERANCIA GENERAL ±0.2					
ANGULO ±1°			HOJA	NUMERO PLANO	REV
			A4	04-02-05-100	0
			ESCALA: 1:1	PESO:	HOJA 1 DE 1

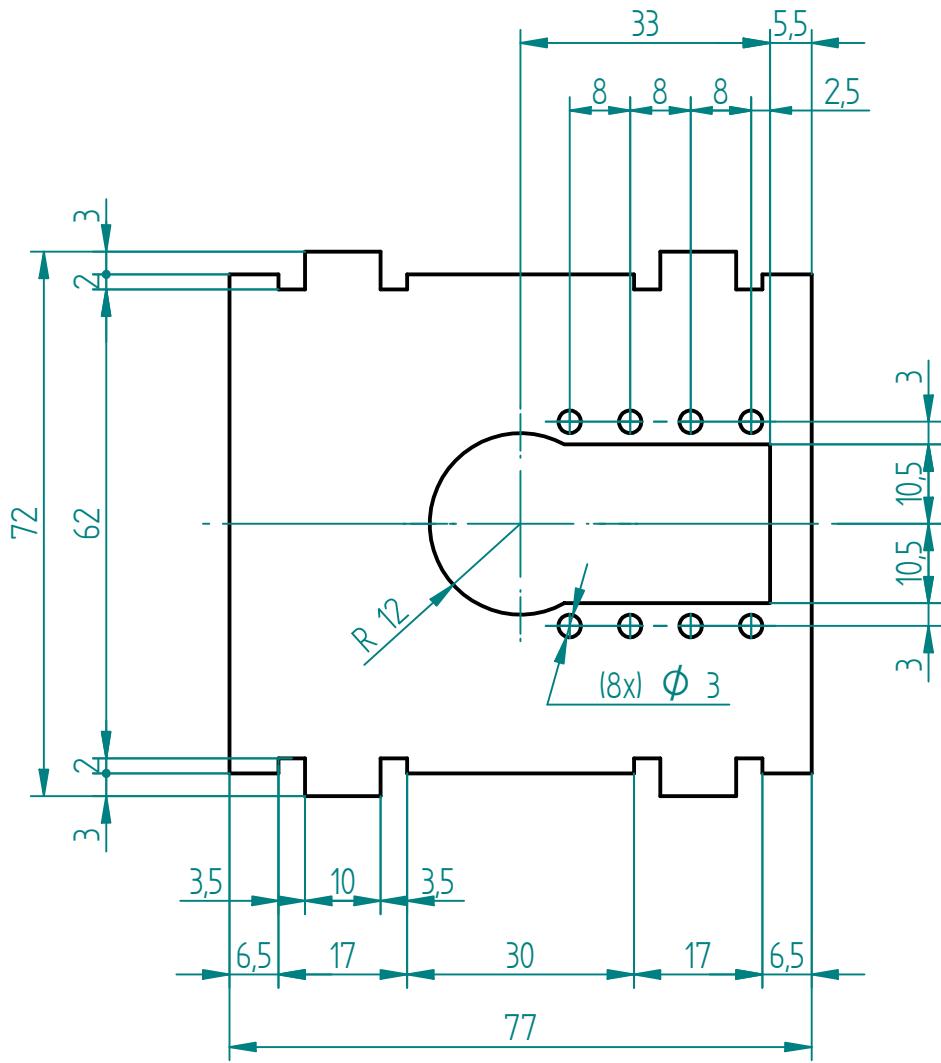


## SECTION A-A



NOMBRE	M.I.G	FECHA	Proyecto Eurobot 2015		
DIBUJADO		28-02-15			
MATERIAL	TRESPA				
ESPESOR	10 MM				
CANTIDAD	1 S/D + 1 M/C	TITULO			BASE CARRO SERVO DERECHA MIR 1
DIMENSIONES ACOTADAS EN MILIMETROS			HOJA	NUMERO PLANO	REV
TOLERANCIA GENERAL ±0.2			A4	04-02-06-101	0
ANGULO ±1°			ESCALA:	1:1	PESO:
			HOJA 1 DE 1		





	NOMBRE	FECHA
DIBUJADO	M.I.G	18-01-15
MATERIAL	TRESPA	
ESPESOR	3 MM	
CANTIDAD	2 UNIDADES	
DIMENSIONES ACOTADAS EN MILIMETROS		
TOLERANCIA GENERAL $\pm 0.2$		
ANGULO $\pm 1^\circ$		

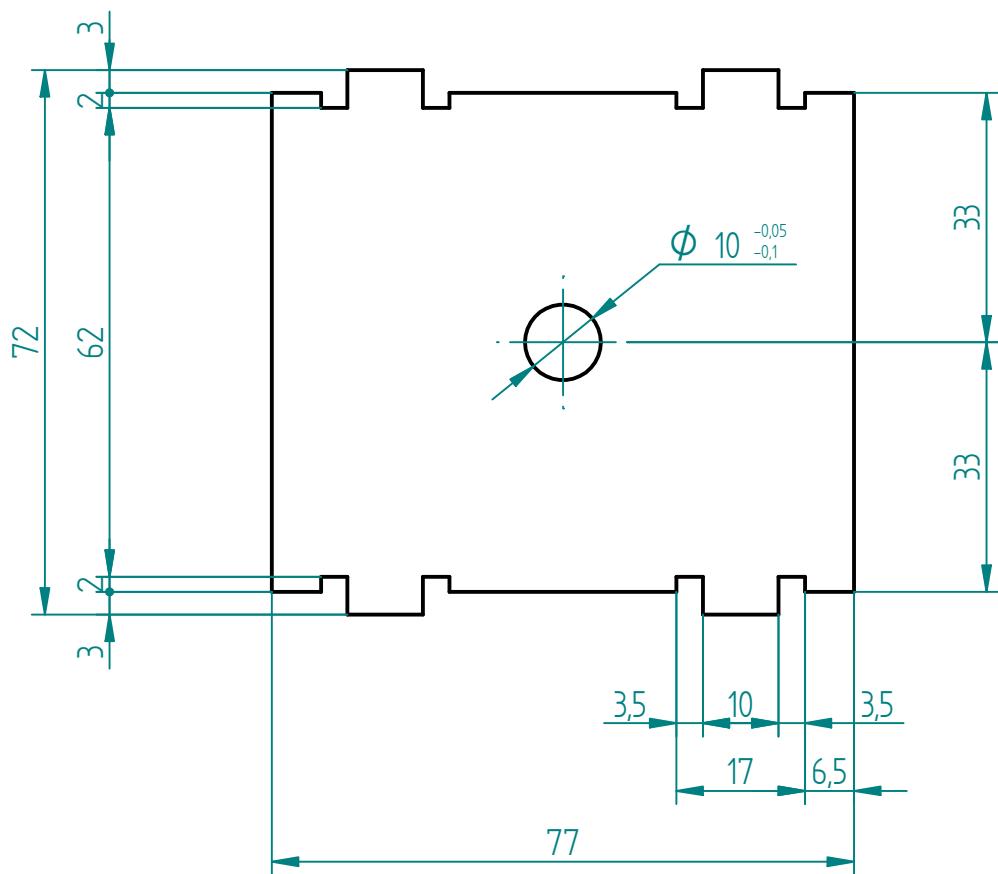
Proyecto Eurobot 2015

TITULO

SOPORTE SERVO CARRO SERVO

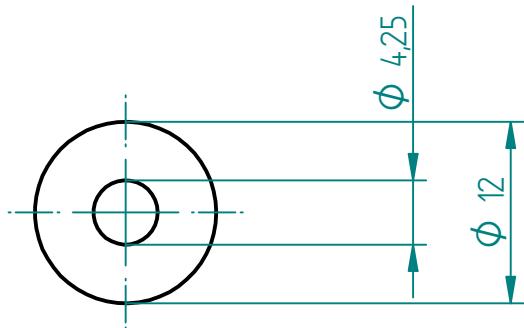
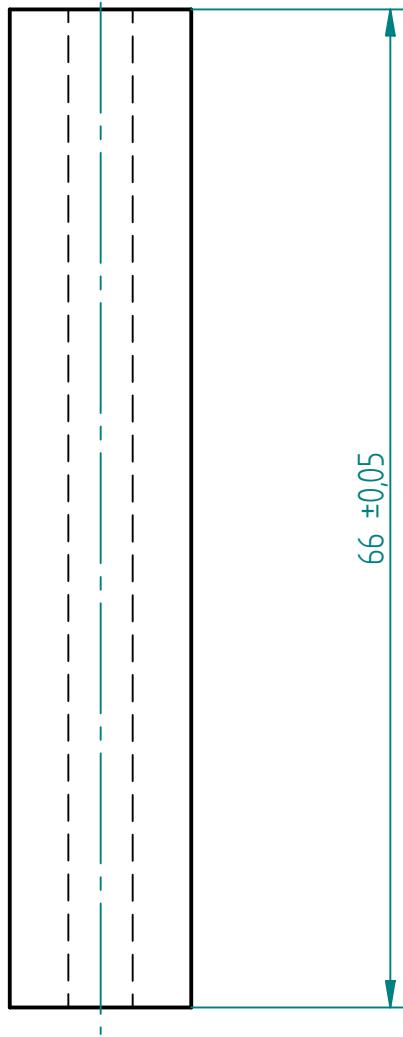
HOJA	NUMERO PLANO	REV
A4	04-02-07-102	0
ESCALA: 1:1	PESO:	HOJA 1 DE 1





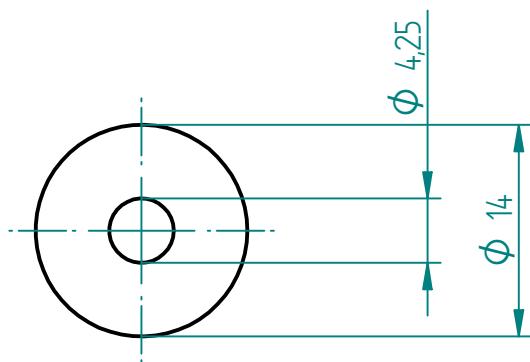
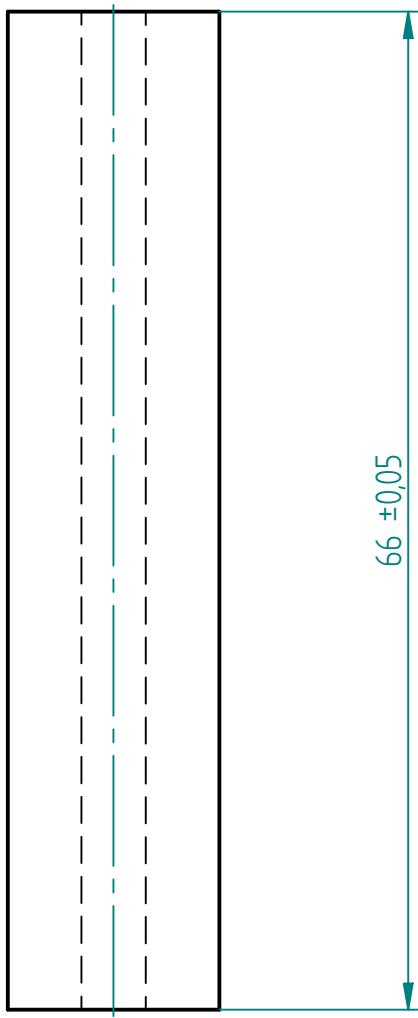
NOMBRE	M.I.G	FECHA	Proyecto Eurobot 2015		
DIBUJADO		18-01-15			
MATERIAL	TRESPA				
ESPESOR	3 MM				
CANTIDAD	2 UNIDADES		TITULO		
DIMENSIONES ACOTADAS EN MILIMETROS			SOPORTE EJE SERVO CARRO SERVO		
TOLERANCIA GENERAL ±0.2			HOJA	NUMERO PLANO	REV
ANGULO ±1°			A4	04-02-08103	0
			ESCALA:	PESO:	HOJA 1 DE 1





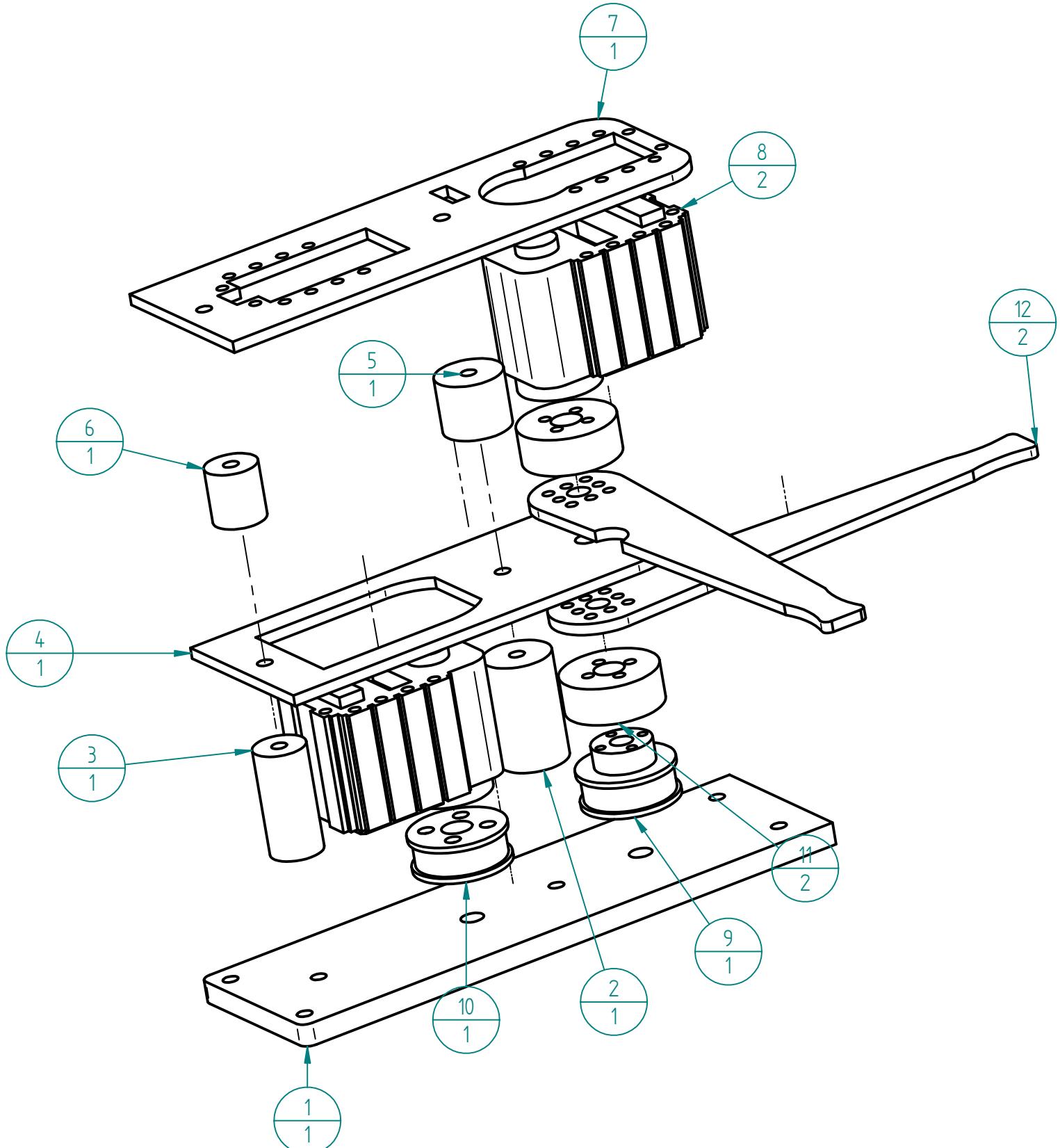
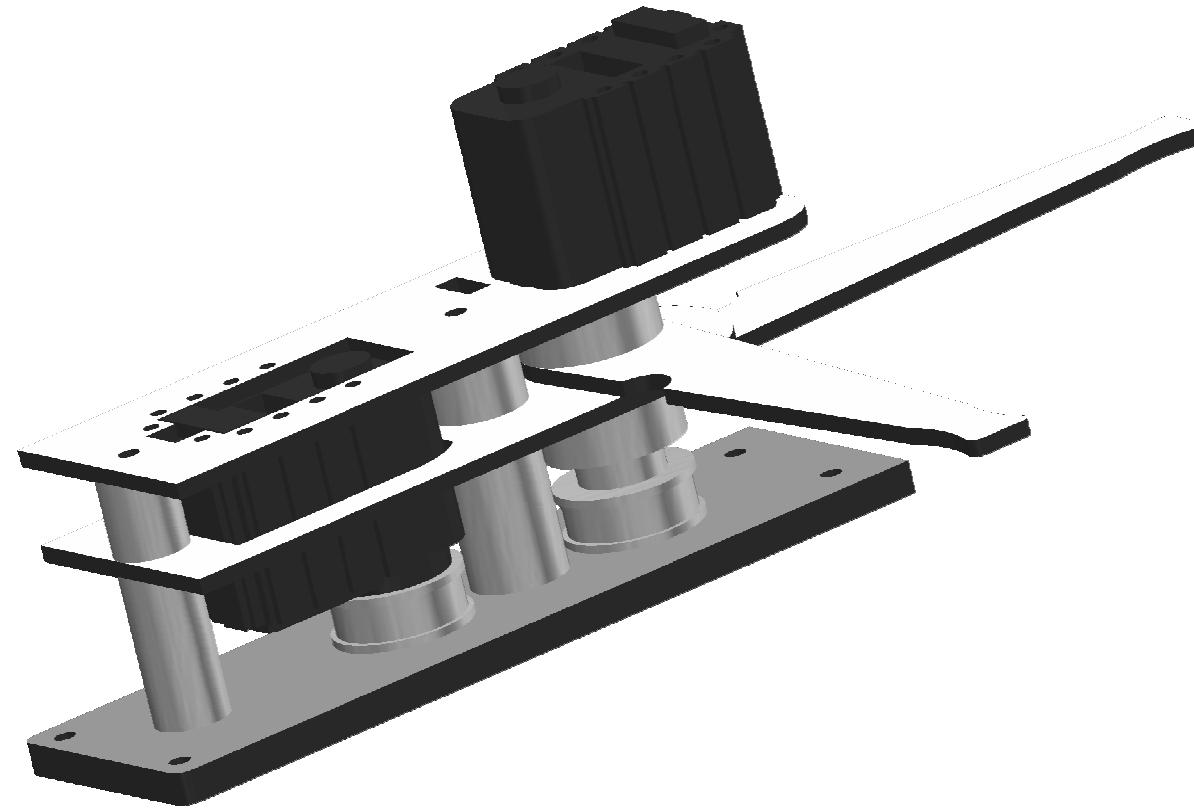
NOMBRE		FECHA	Proyecto Eurobot 2015		
DIBUJADO	M.I.G	18-01-15			
MATERIAL	ALUMINIO			TITULO	
ESPESOR	$\Phi 12$ MM			SEPARADOR LATERAL CARRO SERVO	
CANTIDAD	4 UNIDADES				
DIMENSIONES ACOTADAS EN MILIMETROS TOLERANCIA GENERAL $\pm 0,2$ ANGULO $\pm 1^\circ$			HOJA	NUMERO PLANO	REV
			A4	04-02-09-104	0
			ESCALA:	PESO:	HOJA 1 DE 1





NOMBRE	FECHA	Proyecto Eurobot 2015		
DIBUJADO	M.I.G	18-01-15		
MATERIAL	ALUMINIO	TITULO		
ESPESOR	Ø 14 MM	SEPARADOR CENTRAL CARRO SERVO		
CANTIDAD	2 UNIDADES			
DIMENSIONES ACOTADAS EN MILIMETROS		HOJA	NUMERO PLANO	REV
TOLERANCIA GENERAL ±0.2		A4	04-02-10-105	0
ANGULO ±1°		ESCALA:	2:1	PESO:
				HOJA 1 DE 1

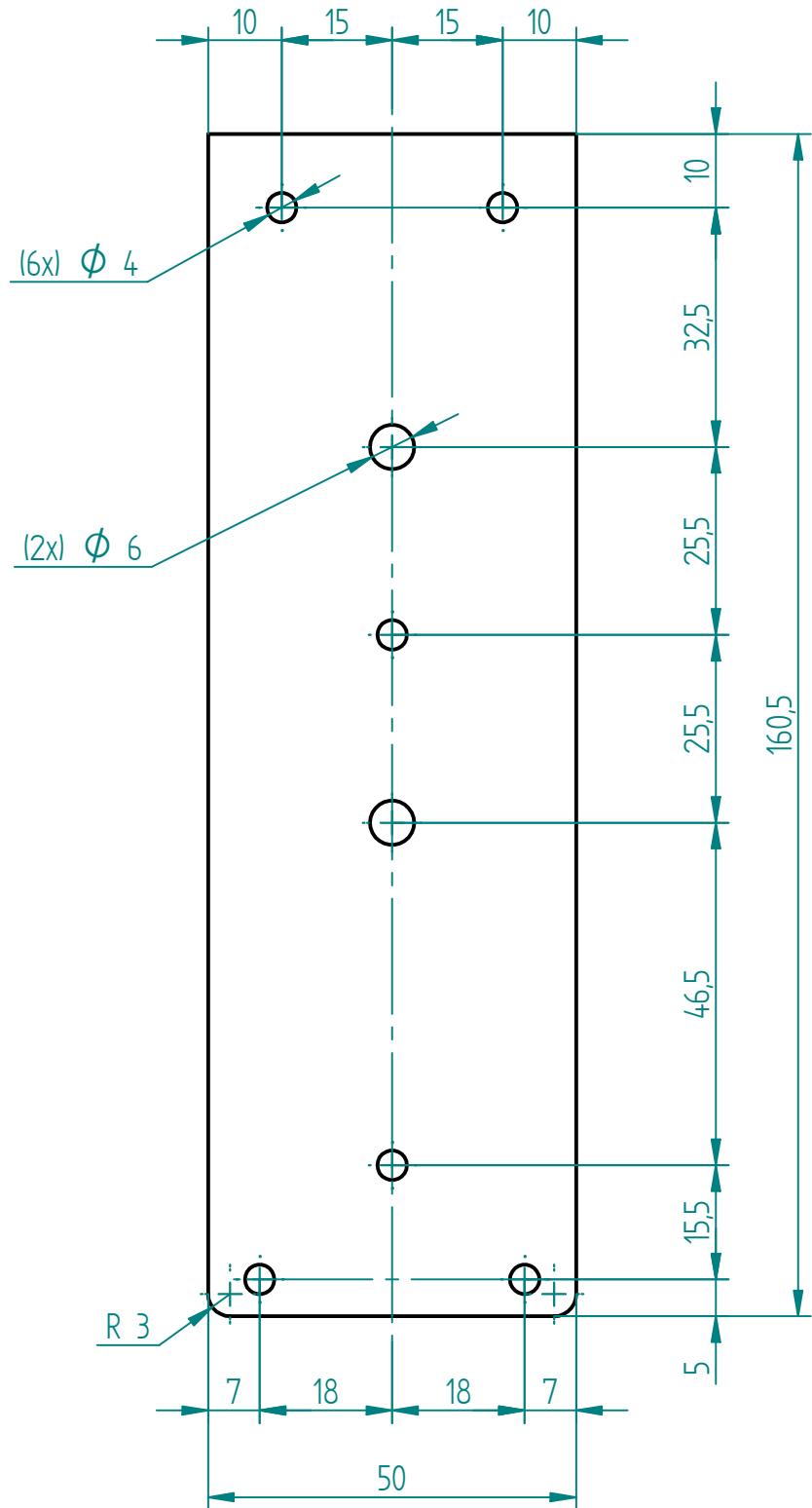




Item	TITULO PLANO	N	Material	Masa
1	Inferior Estructura Paletas	1	Epoxy, cast rigid	0,062 kg
2	Separador Estructura DI	1	Aluminum, 5050	0,019 kg
3	Separador Estructura TI	1	Aluminum, 5050	0,011 kg
4	Medio Estructura Paletas	1	Epoxy, cast rigid	0,018 kg
5	Separador Estructura DS	1	Aluminum, 5050	0,010 kg
6	Separador Estructura TS	1	Aluminum, 5050	0,006 kg
7	Superior Estructura	1	Epoxy, cast rigid	0,020 kg
8	Dynamixel Simplificado	2	(None)	0,054 kg
9	POLEA T25 30	1	Aluminum, 1060	0,014 kg
10	POLEA T25 30 MECANIZADA	1	Aluminum, 1060	0,011 kg
11	Brida Paletas	2	Epoxy, cast rigid	0,006 kg
12	Paleta	2	Epoxy, cast rigid	0,010 kg

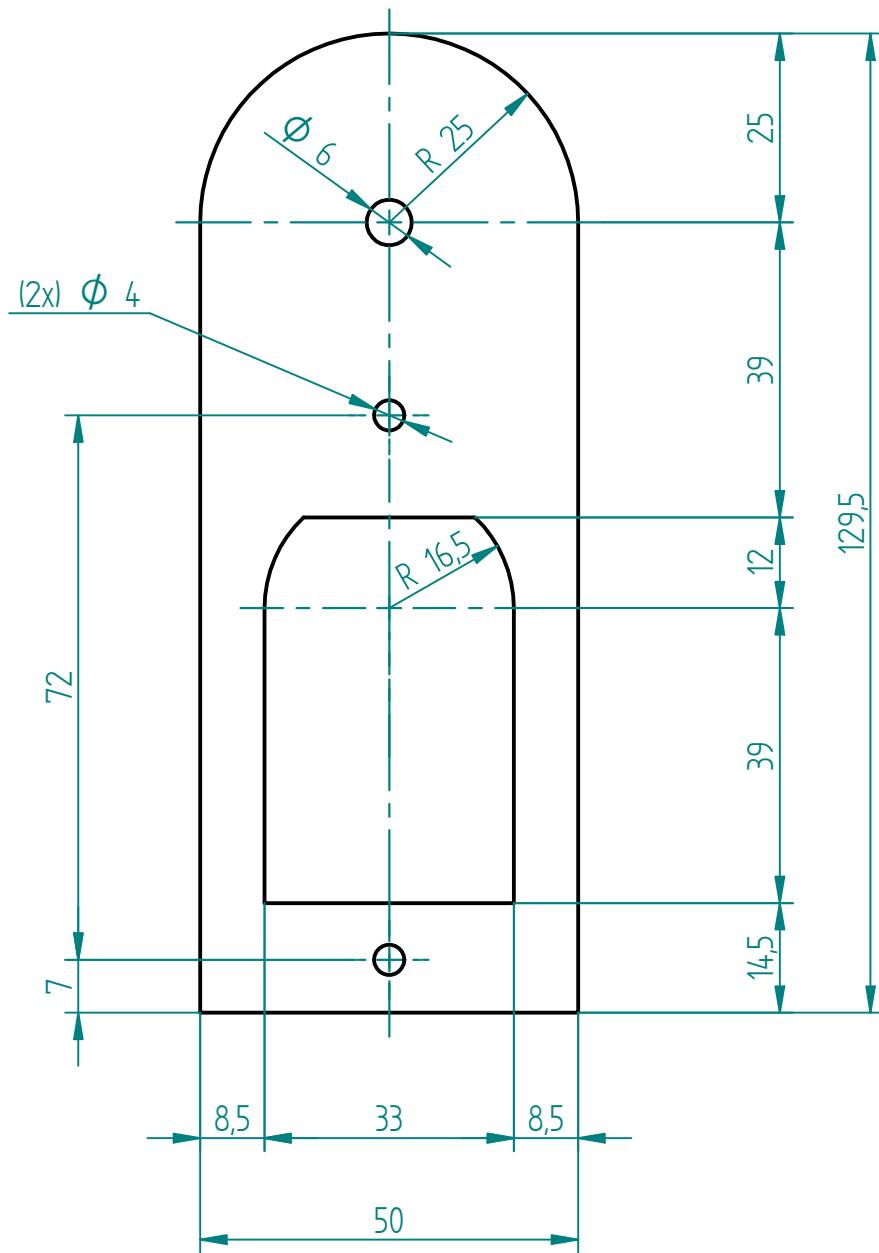
NOMBRE	FECHA	Proyecto Eurobot 2015	
DIBUJADO	M.I.G	15-01-15	
MATERIAL	.		
ESPESOR	.		
CANTIDAD	1 UNIDADES		
DIMENSIONES ACOTADAS EN MILIMETROS		HOJA	NUMERO PLANO
TOLERANCIA GENERAL ±0.2		A3	05-00-00-E01
ANGULO ±1°		REV 0	
ESCALA:		PESO:	HOJA 1 DE 1





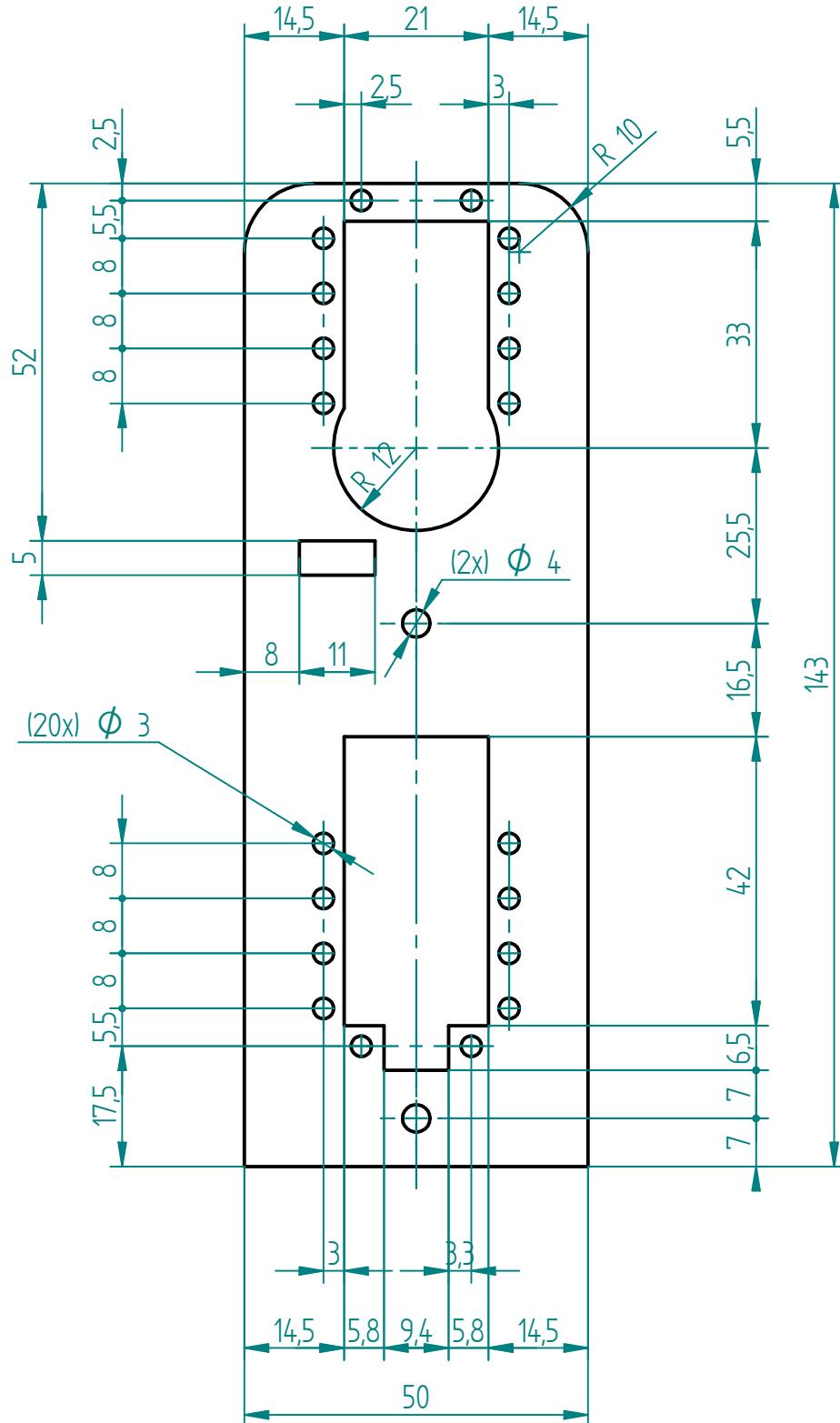
NOMBRE	M.I.G	FECHA	Proyecto Eurobot 2015		
DIBUJADO					
MATERIAL	TRESPA		TITULO		
ESPESOR	6 MM		ESTRUCTURA PALETAS INFERIOR		
CANTIDAD	1 UNIDADES				
DIMENSIONES ACOTADAS EN MILIMETROS TOLERANCIA GENERAL ±0.2 ANGULO ±1°			HOJA	NUMERO PLANO	REV
			A4	05-00-01-002	0
			ESCALA:	PESO:	HOJA 1 DE 1





NOMBRE	M.I.G	FECHA	Proyecto Eurobot 2015		
DIBUJADO					
MATERIAL	TRESPA		TITULO		
ESPESOR	3 MM		ESTRUCTURA PALETAS MEDIO		
CANTIDAD	1 UNIDADES				
DIMENSIONES ACOTADAS EN MILIMETROS			HOJA	NUMERO PLANO	REV
TOLERANCIA GENERAL $\pm 0.2$			A4	05-00-02-034	0
ANGULO $\pm 1^\circ$			ESCALA: 1:1	PESO:	HOJA 1 DE 1





	NOMBRE	FECHA
DIBUJADO	M.I.G	15-01-15
MATERIAL	TRESPA	
ESPESOR	3 MM	
CANTIDAD	1 UNIDADES	
DIMENSIONES ACOTADAS EN MILIMETROS		
TOLERANCIA GENERAL $\pm 0.2$		
ANGULO $\pm 1^\circ$		

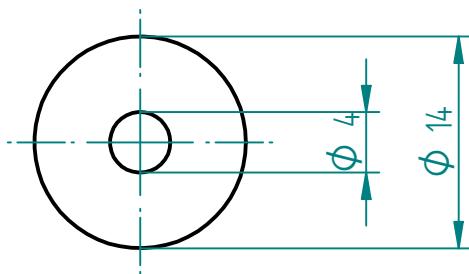
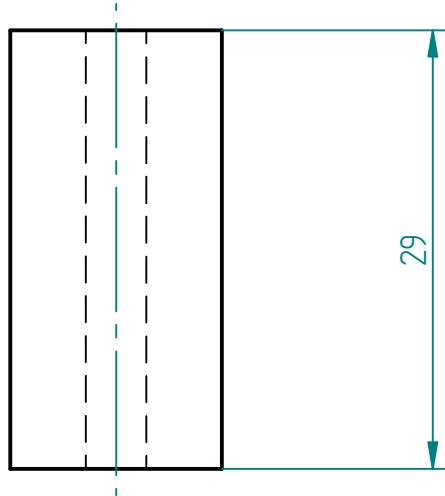
Proyecto Eurobot 2015

TITULO

ESTRUCTURA PALETAS SUPERIOR

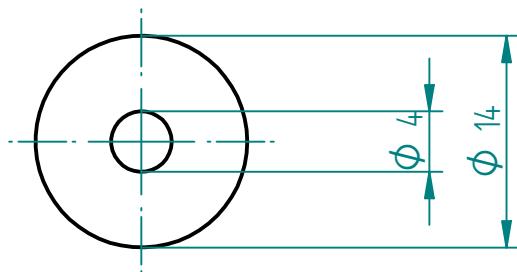
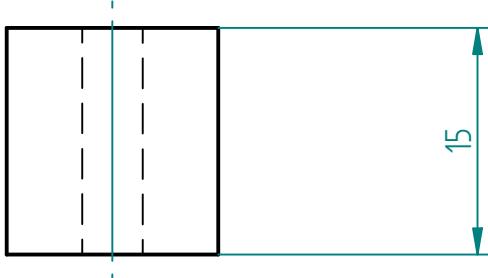
HOJA	NUMERO PLANO	
A4	05-00-03-035	
REV 0		
ESCALA: 1:1	PESO:	HOJA 1 DE 1





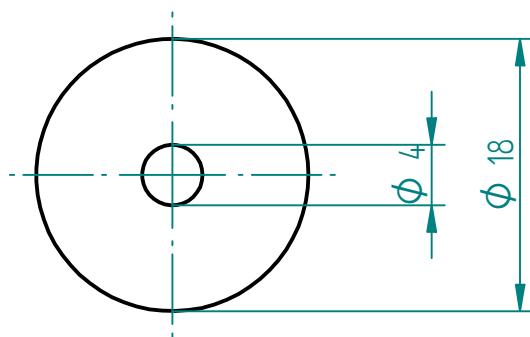
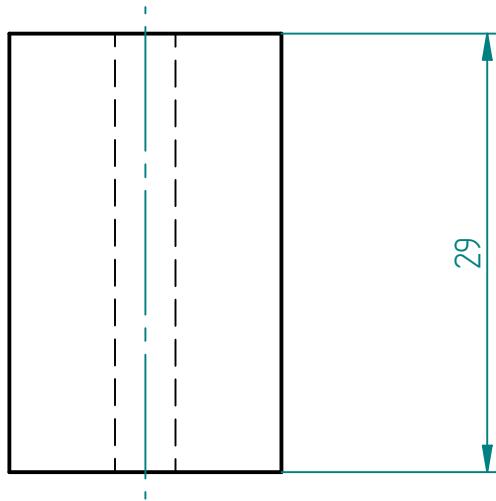
NOMBRE	FECHA	Proyecto Eurobot 2015		
DIBUJADO	M.I.G	15-01-15		
MATERIAL	ALUMINIO DURAL			TITULO
ESPESOR				SEPARADOR ESTRUCTURAS TI
CANTIDAD	1 UNIDADES			
DIMENSIONES ACOTADAS EN MILIMETROS		HOJA	NUMERO PLANO	REV
TOLERANCIA GENERAL $\pm 0.2$		A4	05-00-04-056	0
ANGULO $\pm 1^\circ$		ESCALA: 2:1	PESO:	HOJA 1 DE 1





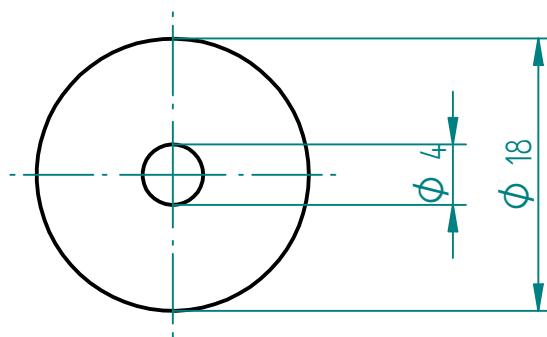
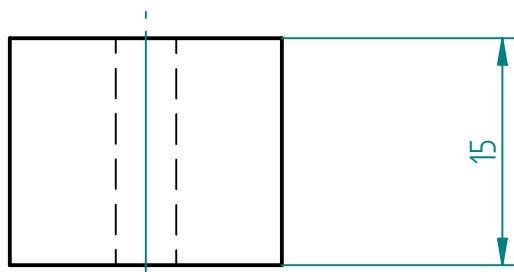
NOMBRE	FECHA	Proyecto Eurobot 2015		
DIBUJADO	M.I.G	15-01-15		
MATERIAL	ALUMINIO DURAL			TITULO
ESPESOR				SEPARADOR ESTRUCTURA TS
CANTIDAD	1 UNIDADES			
DIMENSIONES ACOTADAS EN MILIMETROS		HOJA	NUMERO PLANO	REV
TOLERANCIA GENERAL $\pm 0.2$		A4	05-00-05-057	0
ANGULO $\pm 1^\circ$		ESCALA: 2:1	PESO:	HOJA 1 DE 1





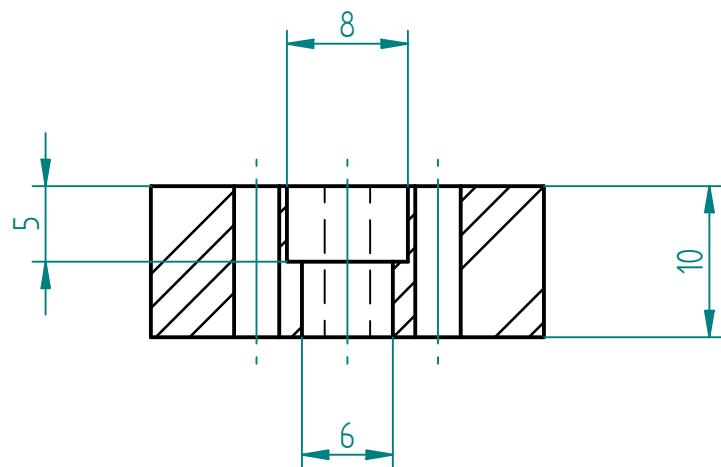
NOMBRE	FECHA	Proyecto Eurobot 2015		
DIBUJADO	M.I.G	15-01-15		
MATERIAL	ALUMINIO DURAL			TITULO
ESPESOR				SEPARADOR ESTRUCTURA DI
CANTIDAD	1 UNIDADES			
DIMENSIONES ACOTADAS EN MILIMETROS		HOJA	NUMERO PLANO	REV
TOLERANCIA GENERAL $\pm 0.2$		A4	05-00-06-058	0
ANGULO $\pm 1^\circ$		ESCALA: 2:1	PESO:	HOJA 1 DE 1



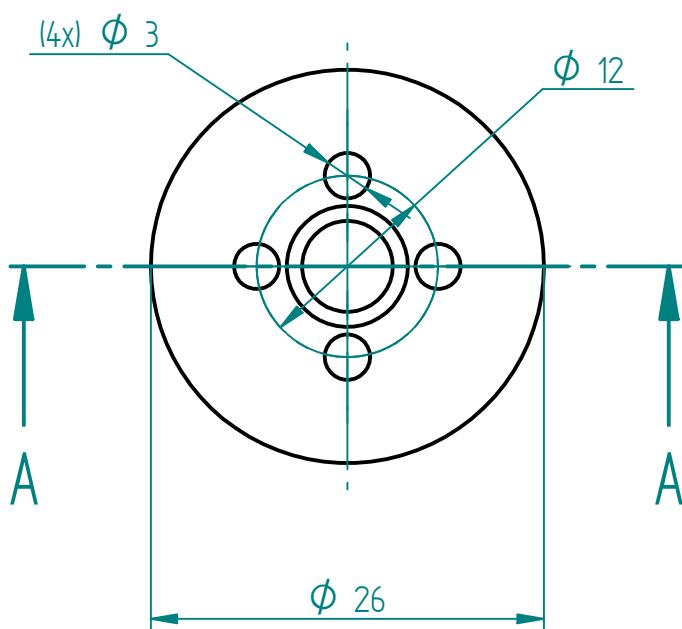


NOMBRE	FECHA	Proyecto Eurobot 2015		
DIBUJADO	M.I.G	15-01-15		
MATERIAL	ALUMINIO DURAL			TITULO
ESPESOR				SEPARADOR ESTRUCTURA DS
CANTIDAD	1 UNIDADES			
DIMENSIONES ACOTADAS EN MILIMETROS		HOJA	NUMERO PLANO	REV
TOLERANCIA GENERAL ±0.2		A4	05-00-07-059	0
ANGULO ±1°		ESCALA:	2:1	PESO:
				HOJA 1 DE 1



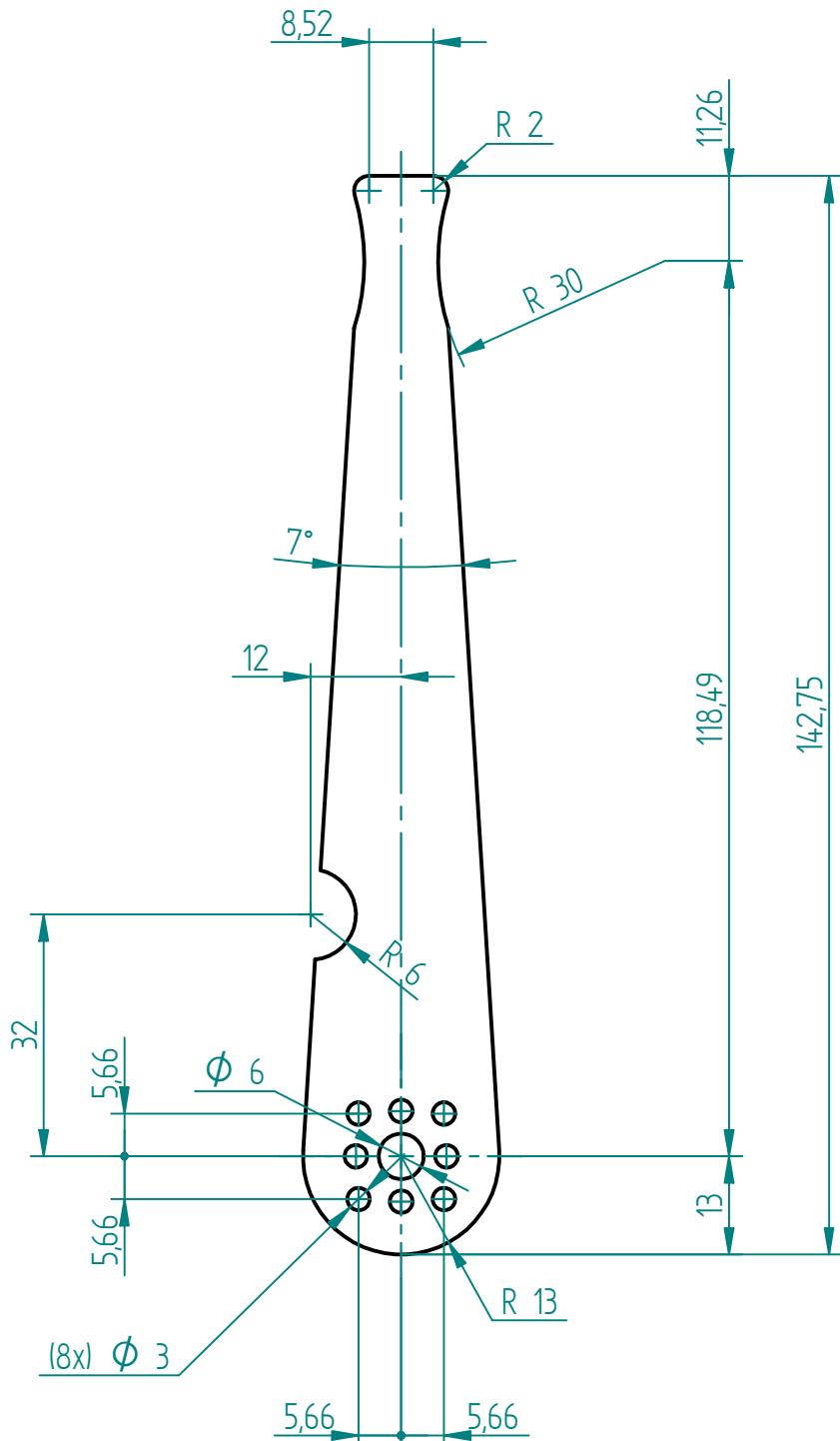


SECTION A-A



NOMBRE	M.I.G	FECHA	Proyecto Eurobot 2015		
DIBUJADO		15-01-15			
MATERIAL	ALUMINIO DURAL			TITULO	
ESPESOR				BRIDA PALETAS	
CANTIDAD	2 UNIDADES				
DIMENSIONES ACOTADAS EN MILIMETROS			HOJA	NUMERO PLANO	REV
TOLERANCIA GENERAL $\pm 0.2$			A4	05-00-08-062	0
ANGULO $\pm 1^\circ$			ESCALA:	PESO:	HOJA 1 DE 1





	NOMBRE	FECHA
DIBUJADO	M.I.G	15-01-15
MATERIAL	TRESPA	
ESPESOR	3 MM	
CANTIDAD	2 UNIDADES	
DIMENSIONES ACOTADAS EN MILIMETROS		HOJA
TOLERANCIA GENERAL $\pm 0.2$		A4
ANGULO $\pm 1^\circ$		NUMERO PLANO
		05-00-09-063
		REV 0

Proyecto Eurobot 2015

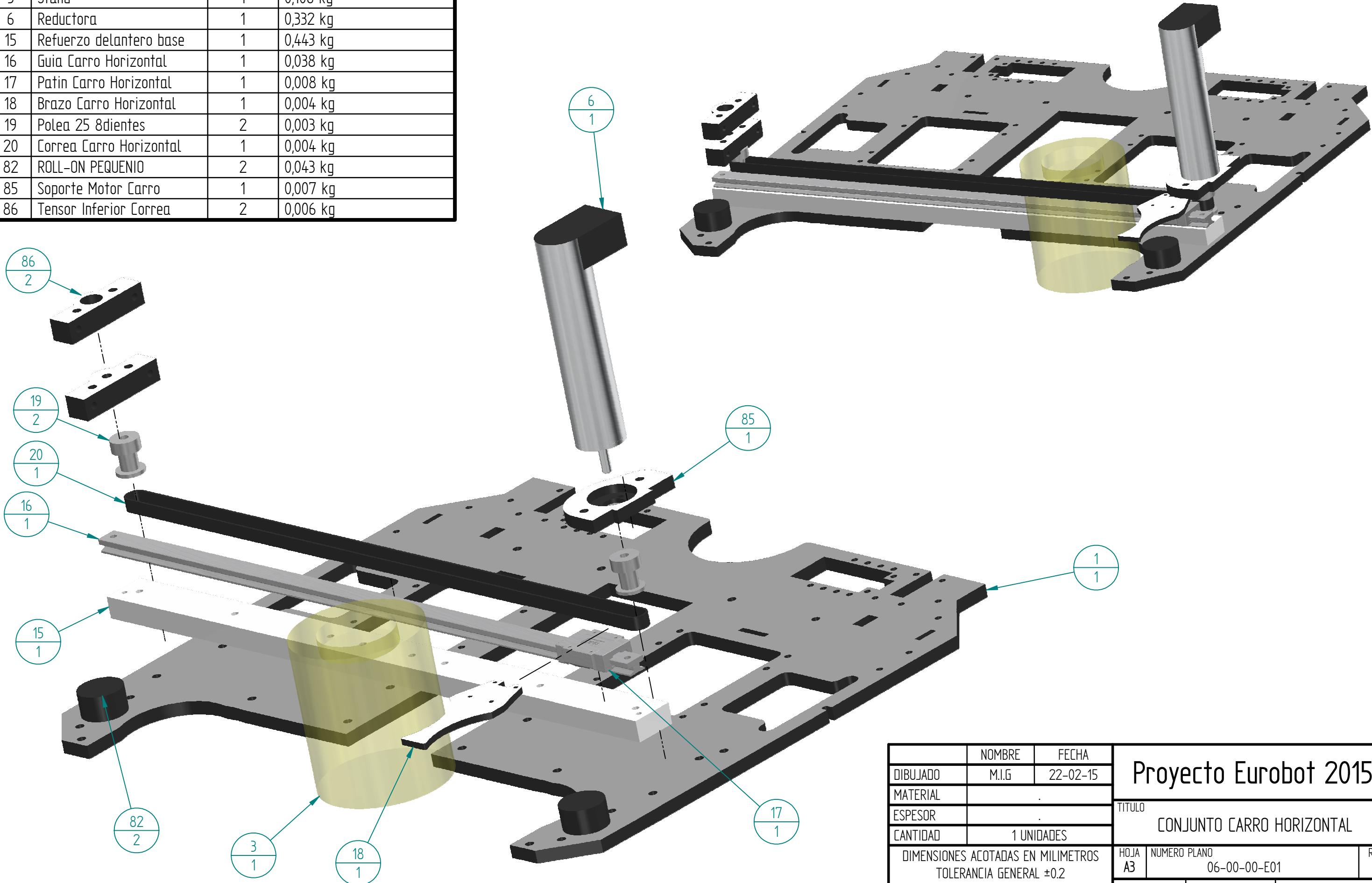
TITULO

PALETA

ESCALA: 1:1	PESO:	HOJA 1 DE 1
-------------	-------	-------------

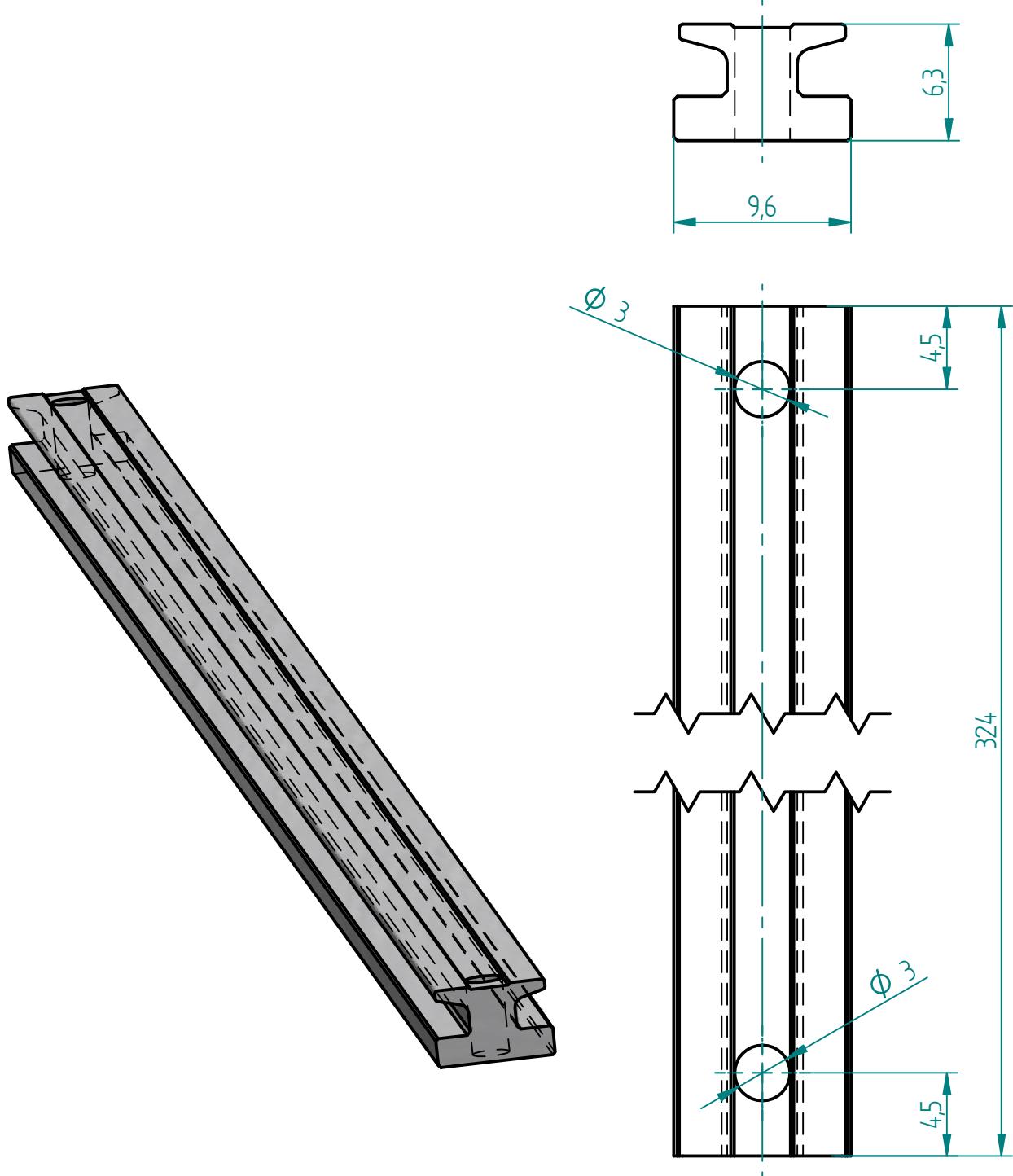


Item	Plano	N.	Masa
1	Base 1	1	0,403 kg
3	Stand	1	0,106 kg
6	Reductora	1	0,332 kg
15	Refuerzo delantero base	1	0,443 kg
16	Guia Carro Horizontal	1	0,038 kg
17	Patin Carro Horizontal	1	0,008 kg
18	Brazo Carro Horizontal	1	0,004 kg
19	Polea 25 8dientes	2	0,003 kg
20	Correa Carro Horizontal	1	0,004 kg
82	ROLL-ON PEQUEÑO	2	0,043 kg
85	Soporte Motor Carro	1	0,007 kg
86	Tensor Inferior Correa	2	0,006 kg



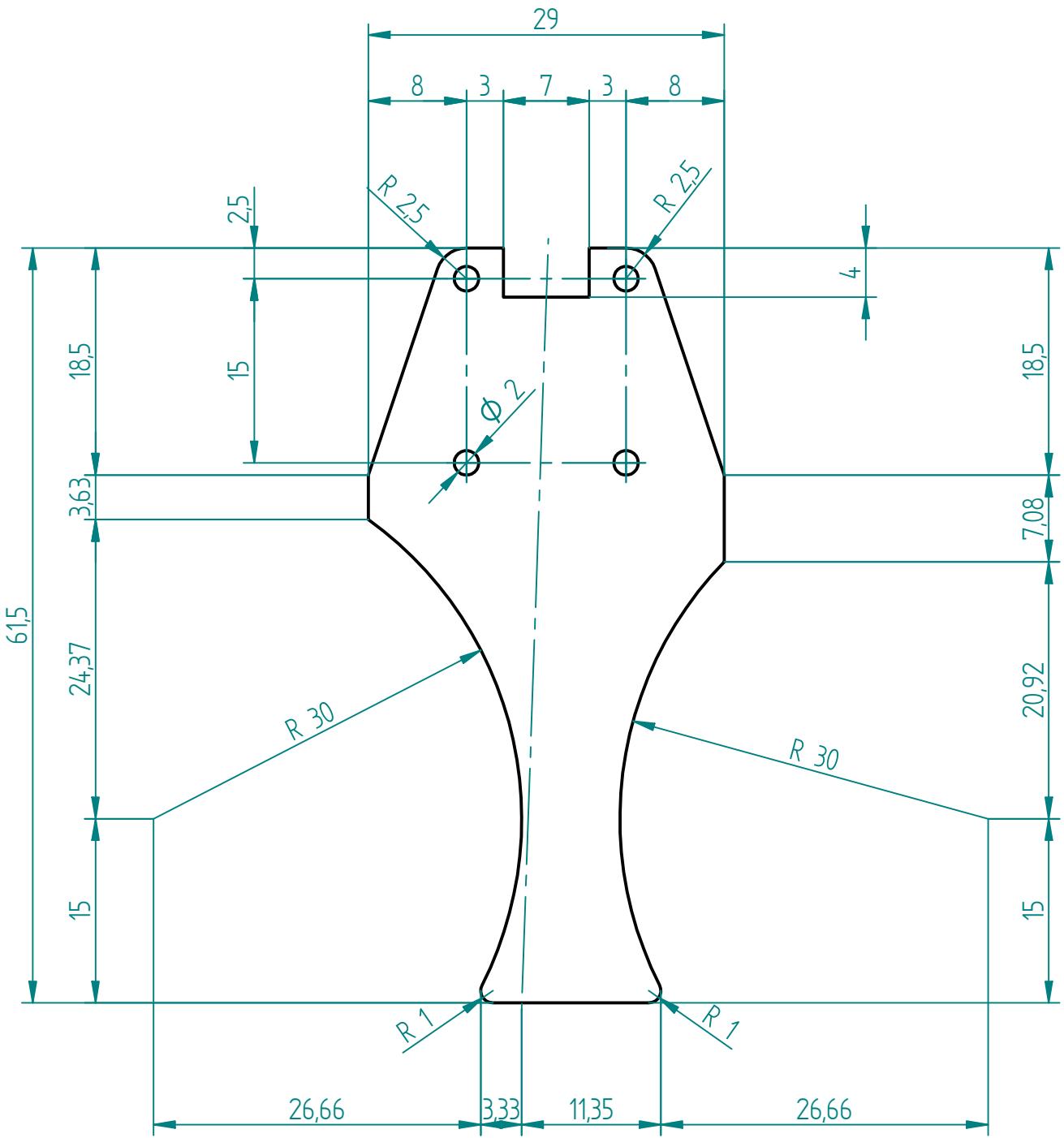
NOMBRE	FECHA	Proyecto Eurobot 2015		
DIBUJADO	M.I.G	22-02-15		
MATERIAL				
ESPESOR				
CANTIDAD	1 UNIDADES			
TITULO				
CONJUNTO CARRO HORIZONTAL				
DIMENSIONES ACOTADAS EN MILIMETROS			REV	
TOLERANCIA GENERAL ±0.2			0	
ANGULO ±1°				
HOJA	NUMERO PLANO			
A3	06-00-00-E01			
ESCALA:		PESO:		HOJA 1 DE 1





NOMBRE		FECHA	Proyecto Eurobot 2015		
DIBUJADO	M.I.G	20-01-15			
MATERIAL	GUIA IGUS TS-04-09			TITULO	
ESPESOR				GUIA CARRO HORIZONTAL	
CANTIDAD	1 UNIDADES				
DIMENSIONES ACOTADAS EN MILIMETROS			HOJA	NUMERO PLANO	REV
TOLERANCIA GENERAL ±0.2			A4	06-00-01-004	0
ANGULO ±1°			ESCALA:	PESO:	HOJA 1 DE 1





	NOMBRE	FECHA
DIBUJADO	M.I.G	20-01-15
MATERIAL	TRESPA	
ESPESOR	3 MM	
CANTIDAD	1 UNIDADES	
DIMENSIONES ACOTADAS EN MILIMETROS		
TOLERANCIA GENERAL $\pm 0.2$		
ANGULO $\pm 1^\circ$		

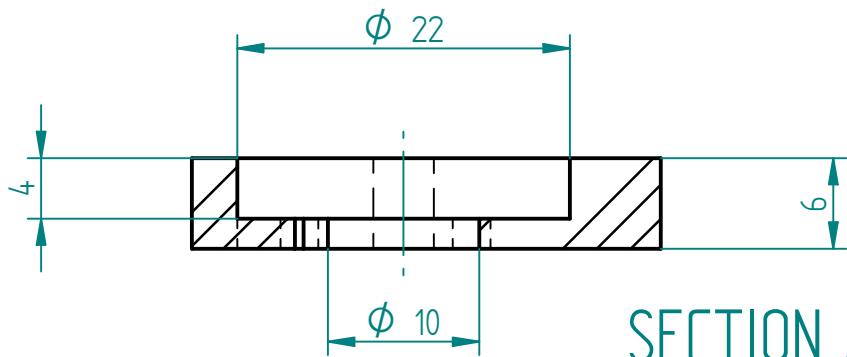
Proyecto Eurobot 2015

TITULO

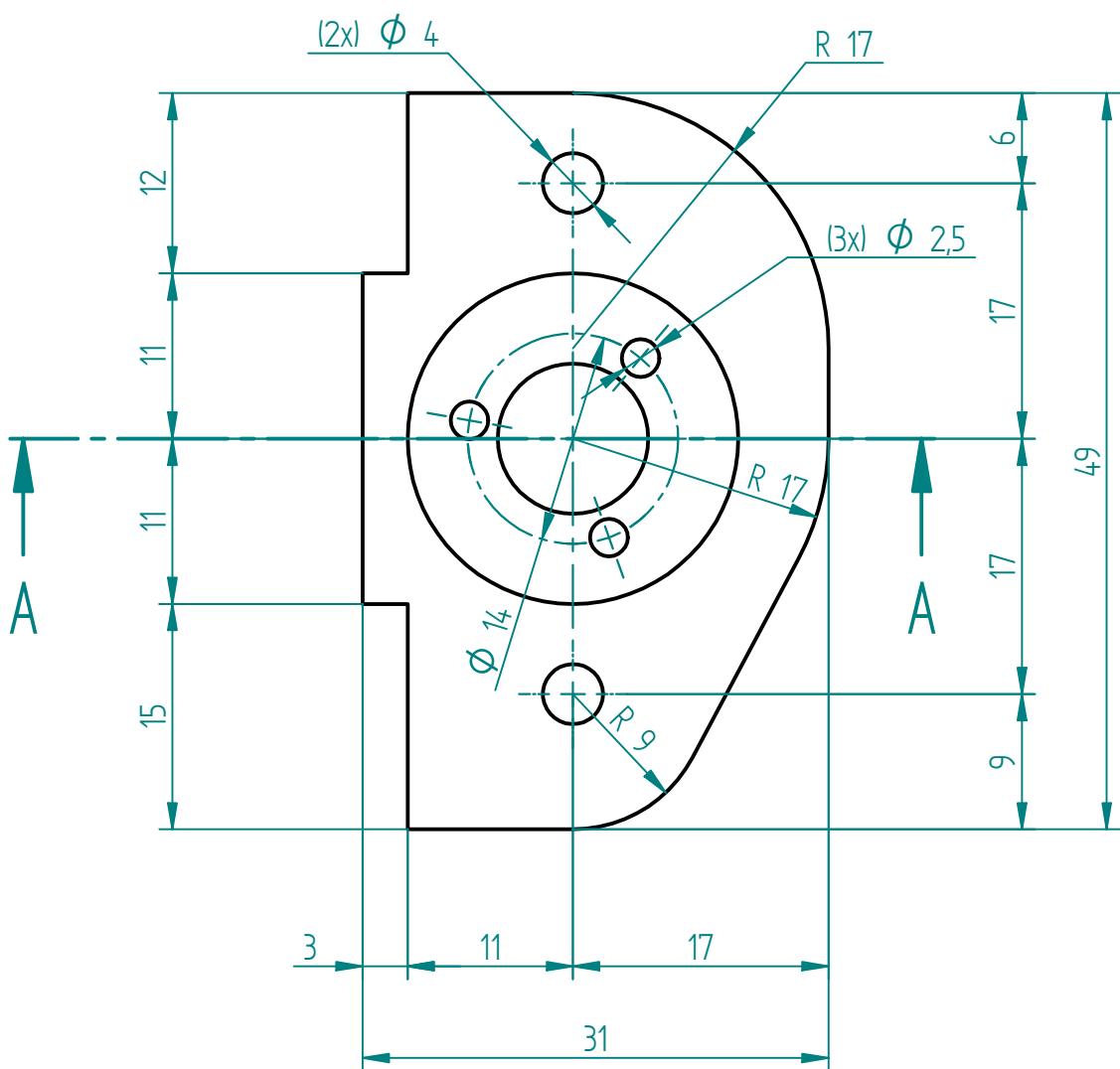
BRAZO CARRO HORIZONTAL

HOJA	NUMERO PLANO	REV
A4	06-00-02-051	0
ESCALA: 2:1	PESO:	HOJA 1 DE 1



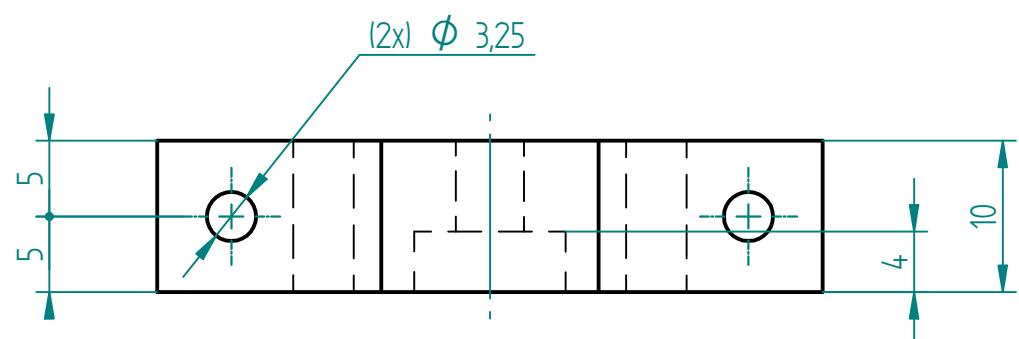
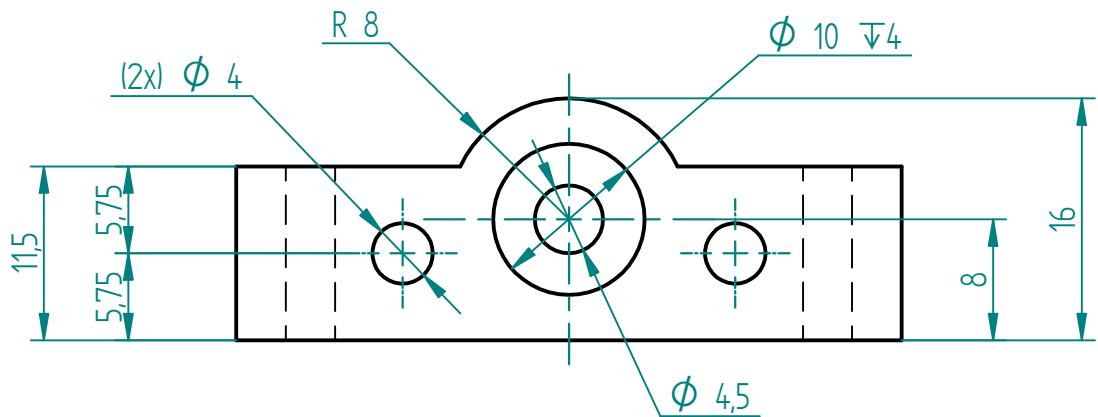


SECTION A-A



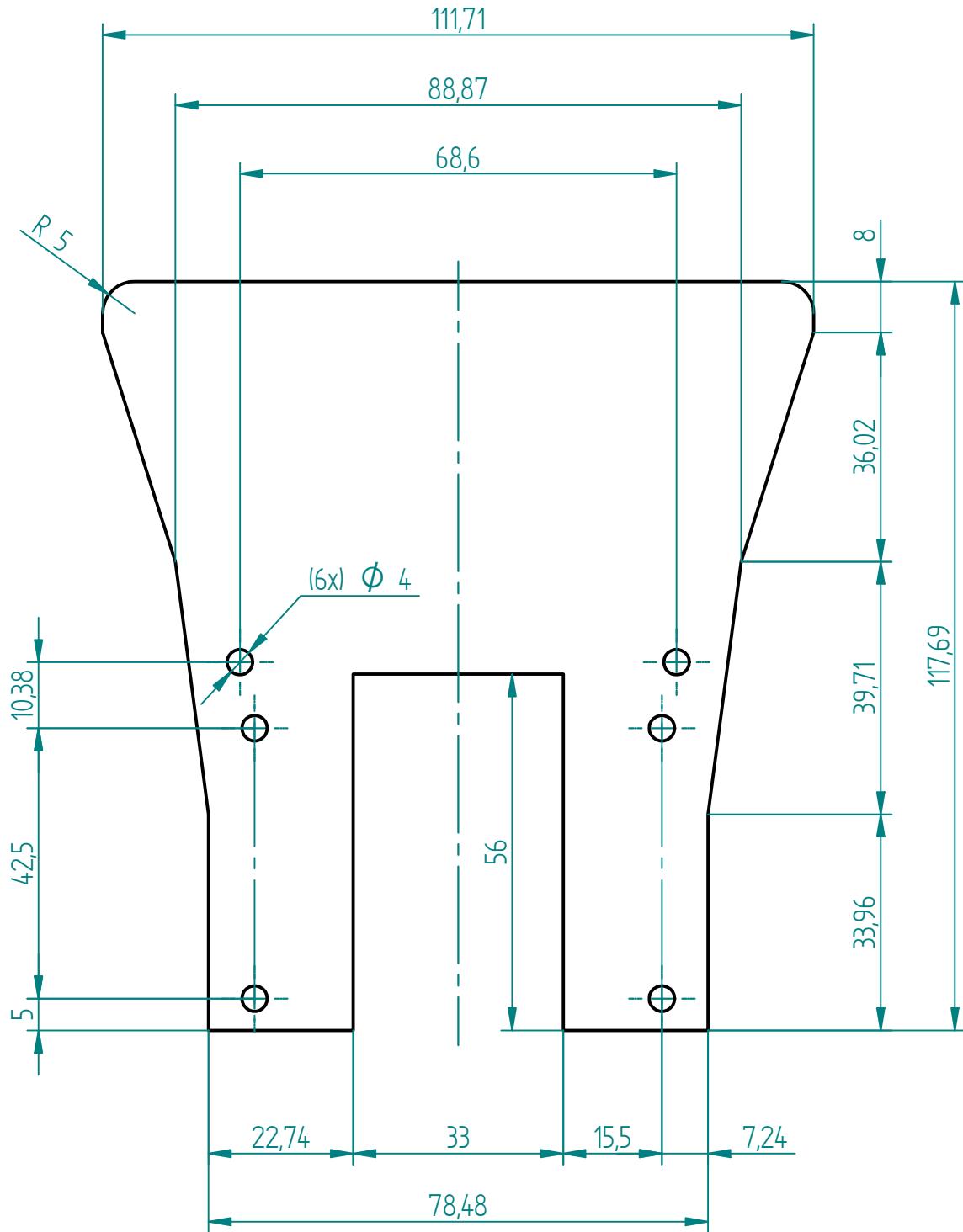
NOMBRE	M.I.G	FECHA	Proyecto Eurobot 2015		
DIBUJADO					
MATERIAL	TRESPA				
ESPESOR	6 MM				
CANTIDAD	1 UNIDADES		TITULO		
DIMENSIONES ACOTADAS EN MILIMETROS			SOPORTE MOTOR CARRO HORIZONTAL		
TOLERANCIA GENERAL ±0.2					
ANGULO ±1°			HOJA	NUMERO PLANO	REV
			A4	06-00-03-067	0
			ESCALA:	PESO:	HOJA 1 DE 1





NOMBRE	M.I.G	FECHA	Proyecto Eurobot 2015		
DIBUJADO					
MATERIAL	TRESPA		TITULO		
ESPESOR	10 MM		TENSOR INFERIOR CORREA		
CANTIDAD	2 UNIDADES				
DIMENSIONES ACOTADAS EN MILIMETROS TOLERANCIA GENERAL ±0.2 ANGULO ±1°			HOJA A4	NUMERO PLANO 06-00-04-070	REV 0
			ESCALA: 2:1	PESO:	HOJA 1 DE 1





	NOMBRE	FECHA
DIBUJADO	M.I.G	21-01-15
MATERIAL	TRESPA	
ESPESOR	6 MM	
CANTIDAD	1 UNIDADES	

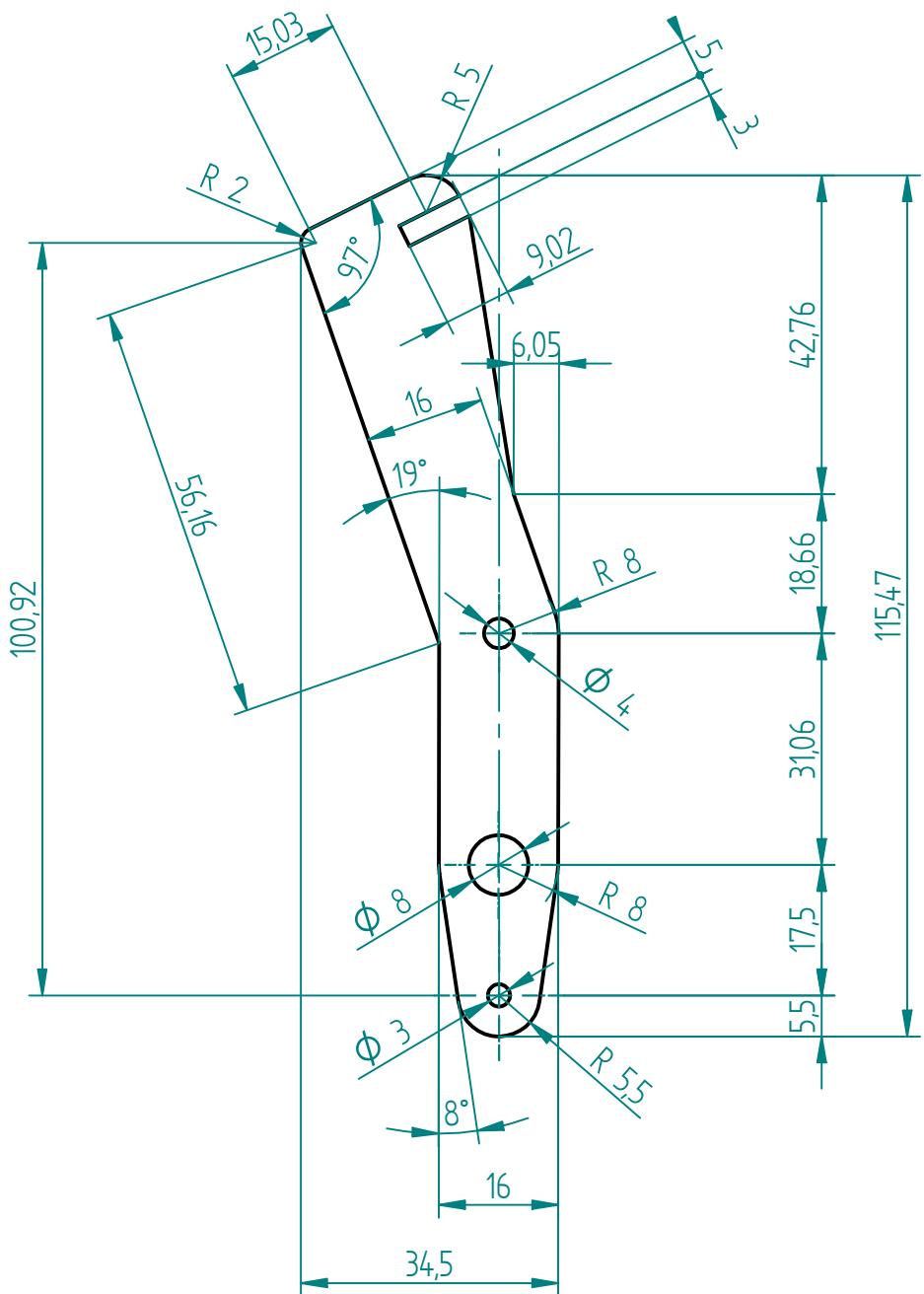
DIMENSIONES ACOTADAS EN MILIMETROS  
TOLERANCIA GENERAL  $\pm 0.2$   
ANGULO  $\pm 1^\circ$

## Proyecto Eurobot 2015

TITULO  
BASE PUERTA STAND

HOJA A4	NUMERO PLANO 07-00-01-020	REV 0
ESCALA: 1:1	PESO:	HOJA 1 DE 1



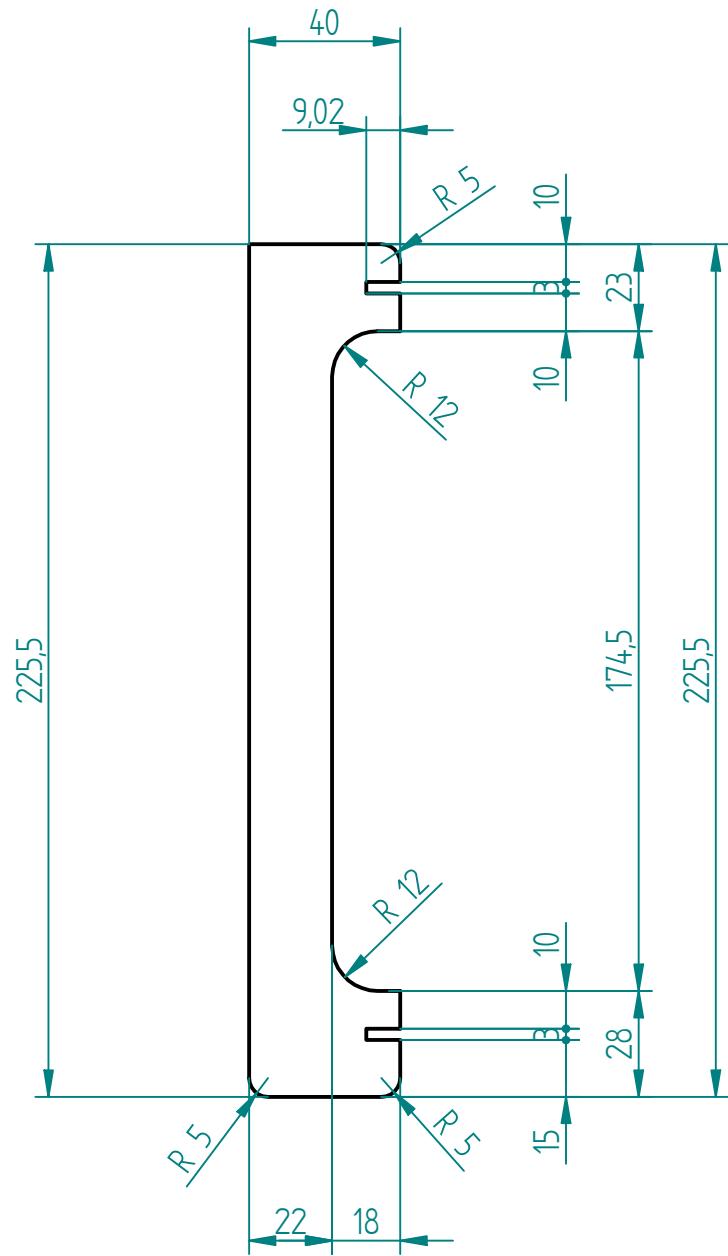


	NOMBRE	FECHA
DIBUJADO	M.I.G	21-01-15
MATERIAL	TRESPA	
ESPESOR	3 MM	
CANTIDAD	4 UNIDADES	
DIMENSIONES ACOTADAS EN MILIMETROS		
TOLERANCIA GENERAL $\pm 0.2$		
ANGULO $\pm 1^\circ$		

Proyecto Eurobot 2015

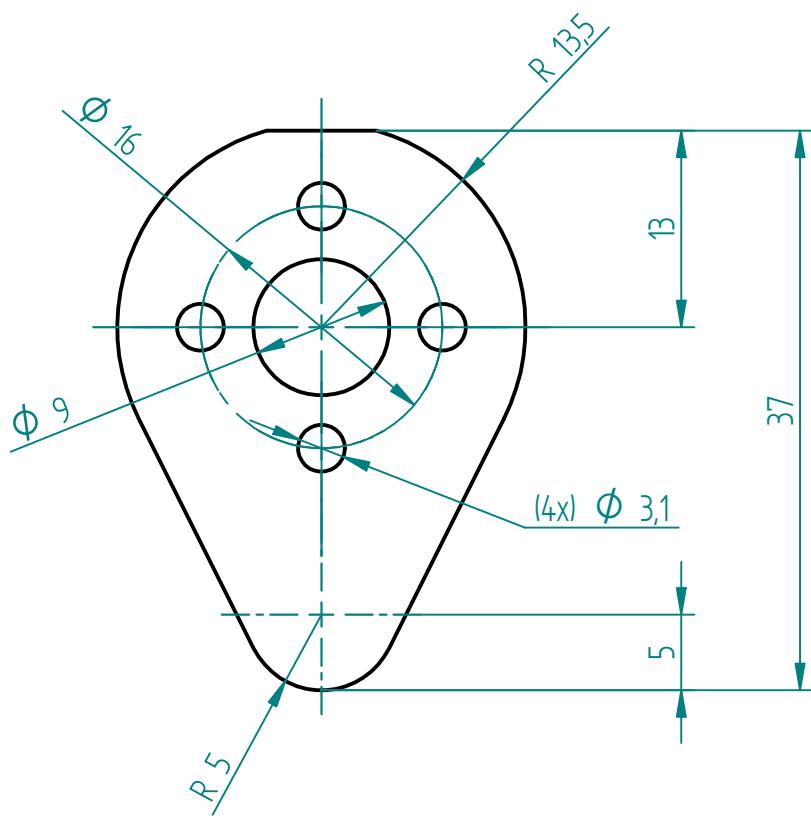
TITULO	BRAZO PUERTA SUPERIOR		
HOJA	NUMERO PLANO	REV	
A4	07-00-02-047	0	
ESCALA: 1:1	PESO:	HOJA 1 DE 1	





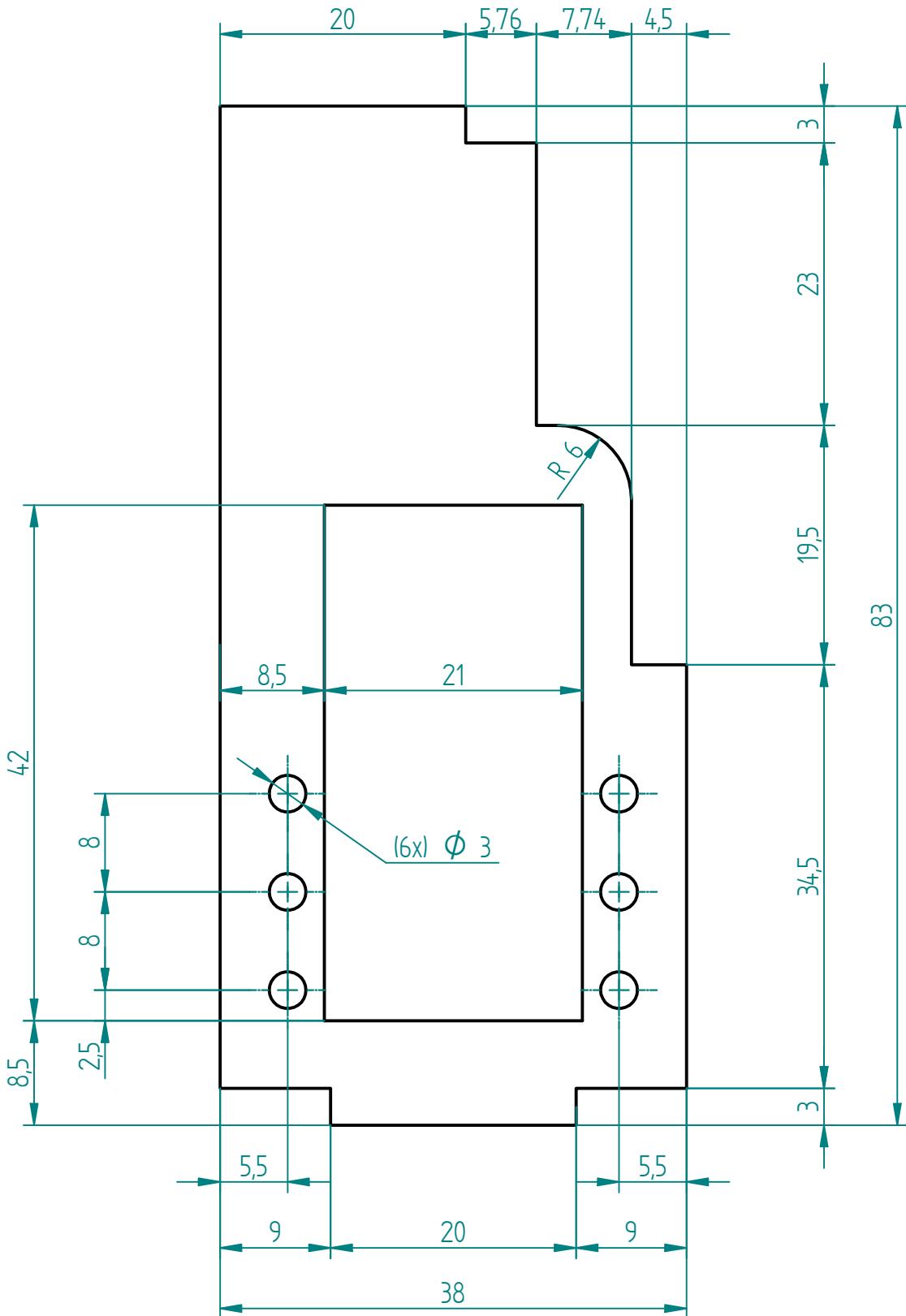
NOMBRE	M.I.G	FECHA	Proyecto Eurobot 2015		
DIBUJADO					
MATERIAL	TRESPA		TITULO		
ESPESOR	3 MM		PUERTA		
CANTIDAD	2 UNIDADES				
DIMENSIONES ACOTADAS EN MILIMETROS TOLERANCIA GENERAL ±0.2 ANGULO ±1°			HOJA A4	NUMERO PLANO 07-00-03-048	REV 0
			ESCALA: 1:2	PESO:	HOJA 1 DE 1





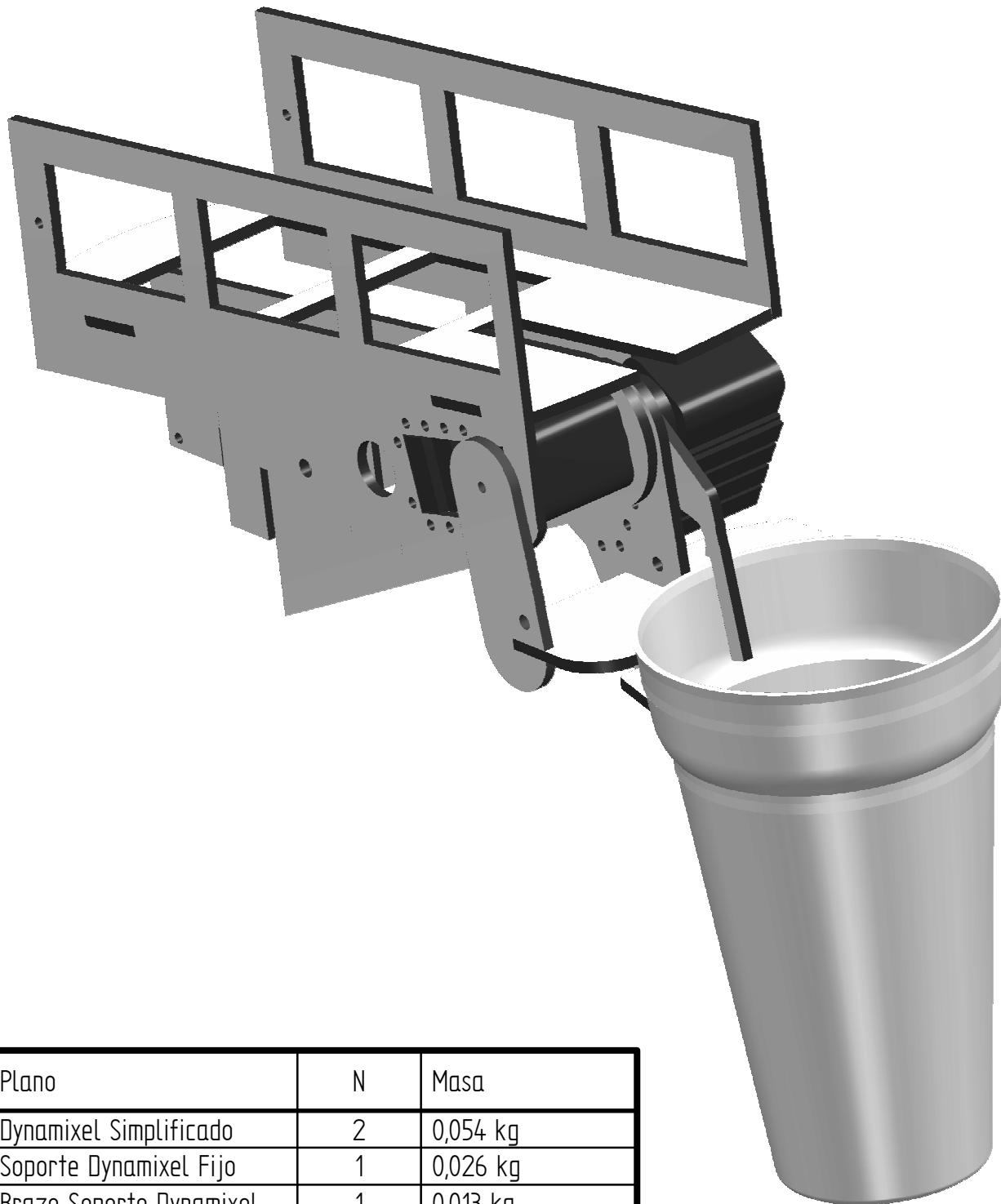
NOMBRE		FECHA	Proyecto Eurobot 2015		
DIBUJADO	M.I.G	21-01-15			
MATERIAL	TRESPA				TITULO
ESPESOR	3 MM				LEVA SERVO PUERTA
CANTIDAD	2 UNIDADES				
DIMENSIONES ACOTADAS EN MILIMETROS			HOJA	NUMERO PLANO	REV
TOLERANCIA GENERAL ±0.2			A4	07-00-04-079	0
ANGULO ±1°			ESCALA:	2.1	PESO:
			HOJA 1 DE 1		



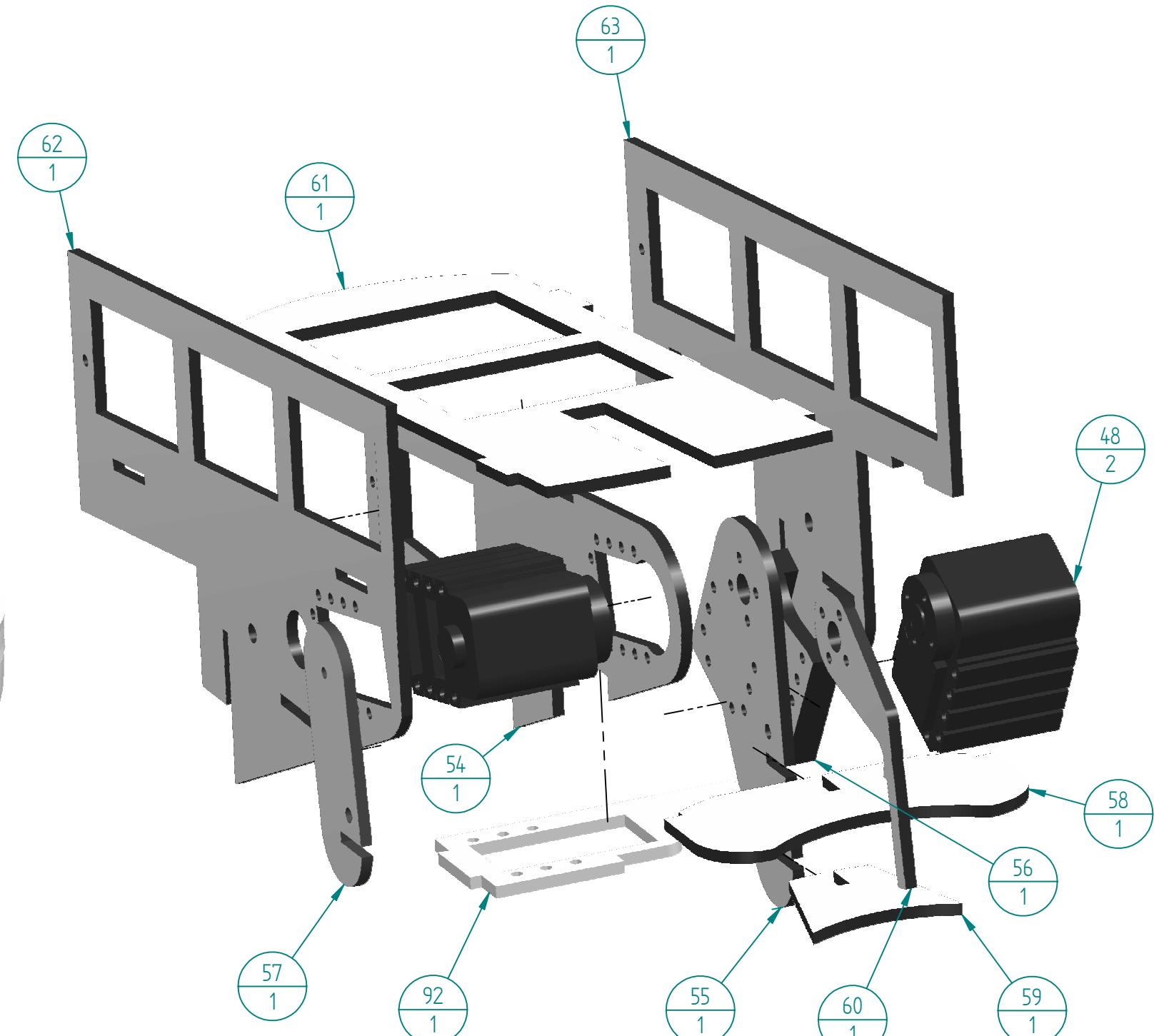


NOMBRE	M.I.G	FECHA	Proyecto Eurobot 2015			
DIBUJADO						
MATERIAL	TRESPA		TITULO			
ESPESOR	3 MM		SOPORTE SERVO INFERIOR			
CANTIDAD	1 UNIDADES					
DIMENSIONES ACOTADAS EN MILIMETROS TOLERANCIA GENERAL ±0.2 ANGULO ±1°			HOJA	NUMERO PLANO	REV	
			A4	07-00-05-200	0	
			ESCALA:	PESO:	HOJA 1 DE 1	



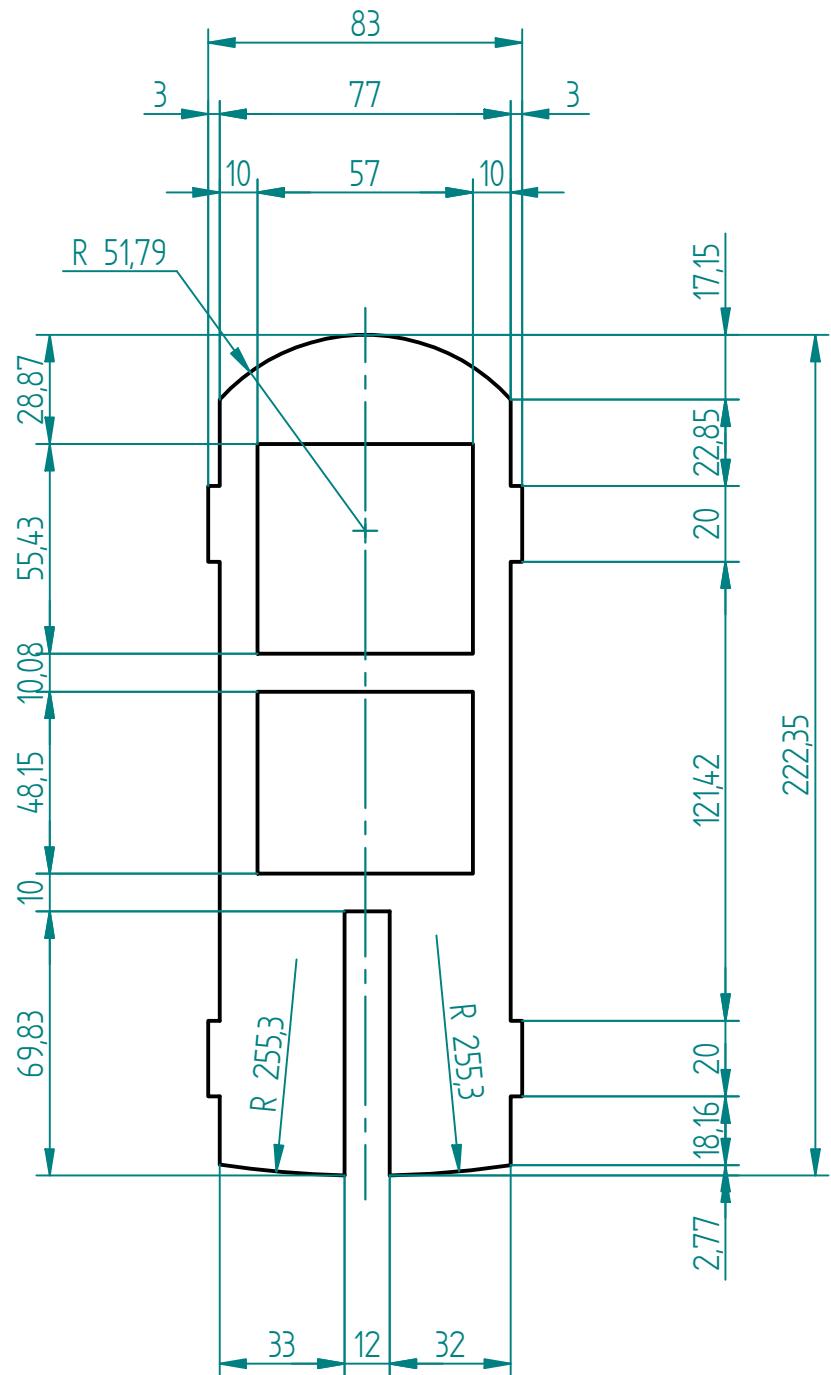


Item	Plano	N	Masa
48	Dynamixel Simplificado	2	0,054 kg
54	Soporte Dynamixel Fijo	1	0,026 kg
55	Brazo Soporte Dynamixel	1	0,013 kg
56	Separador Dynamixel Movil	1	0,008 kg
57	Brazo Estabilizador	1	0,006 kg
58	Apoyo Superior	1	0,011 kg
59	Apoyo Inferior	1	0,004 kg
60	Pinza	1	0,004 kg
61	Rampa Base	1	0,040 kg
62	Soporte Dynamixel Fijo	1	0,047 kg
63	Soporte Dynamixel Fijo	1	0,041 kg
92	Soporte Servo Inferior	1	0,007 kg



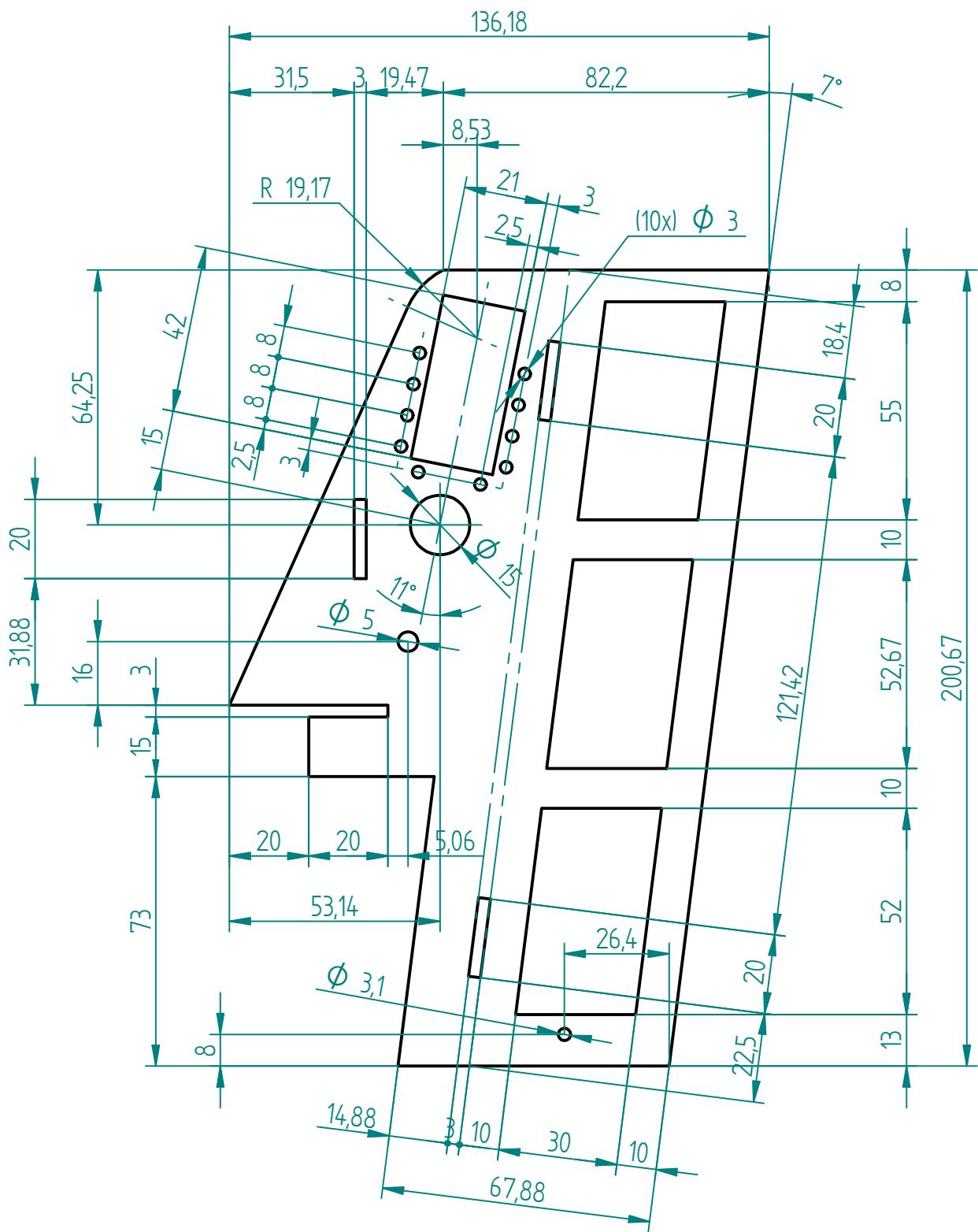
	NOMBRE	FECHA	Proyecto Eurobot 2015			
DIBUJADO	M.I.G	17-02-15				
MATERIAL			TITULO			
ESPESOR			CONJUNTO PINZA VASO DELANTERO			
CANTIDAD	1 UNIDADES					
DIMENSIONES ACOTADAS EN MILIMETROS			HOJA	NUMERO PLANO		
TOLERANCIA GENERAL ±0.2			A3	08-00-00-E01		
ANGULO ±1°			REV	0		
ESCALA:		PESO:	HOJA 1 DE 1			





NOMBRE	M.I.G	FECHA	Proyecto Eurobot 2015			
DIBUJADO						
MATERIAL	TRESPA		TITULO			
ESPESOR	3 MM		RAMPA BASE			
CANTIDAD	1 UNIDADES					
DIMENSIONES ACOTADAS EN MILIMETROS TOLERANCIA GENERAL $\pm 0.2$ ANGULO $\pm 1^\circ$			HOJA	NUMERO PLANO	REV	
			A4	08-00-01-021	0	
			ESCALA: 1:2	PESO:	HOJA 1 DE 1	





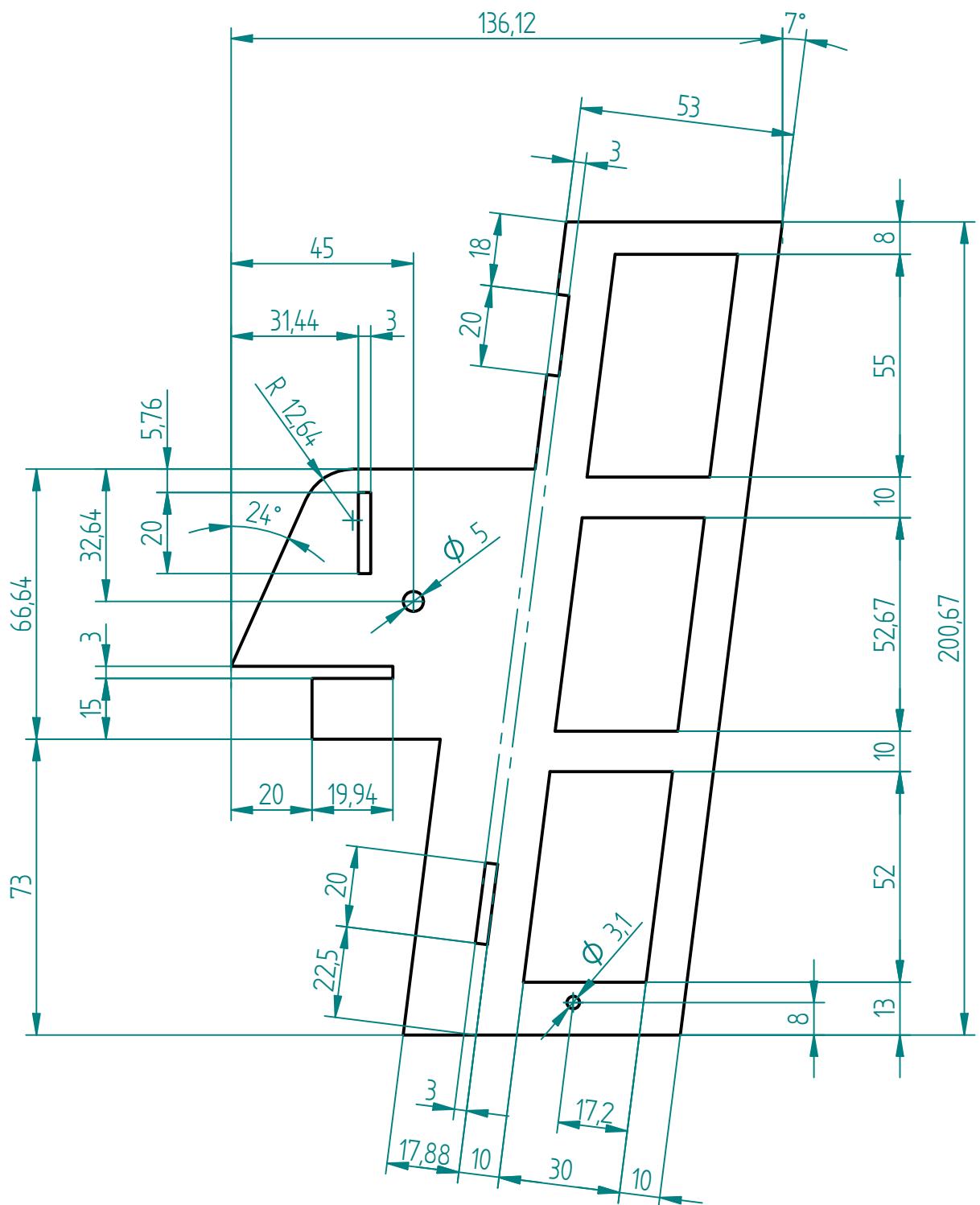
	NOMBRE	FECHA
DIBUJADO	M.I.G	17-02-15
MATERIAL	TRESPA	
ESPESOR	3 MM	
CANTIDAD	1 UNIDADES	

## Proyecto Eurobot 2015

TITULO  
SOPORTE DYNAMIXEL FIJO LATERAL

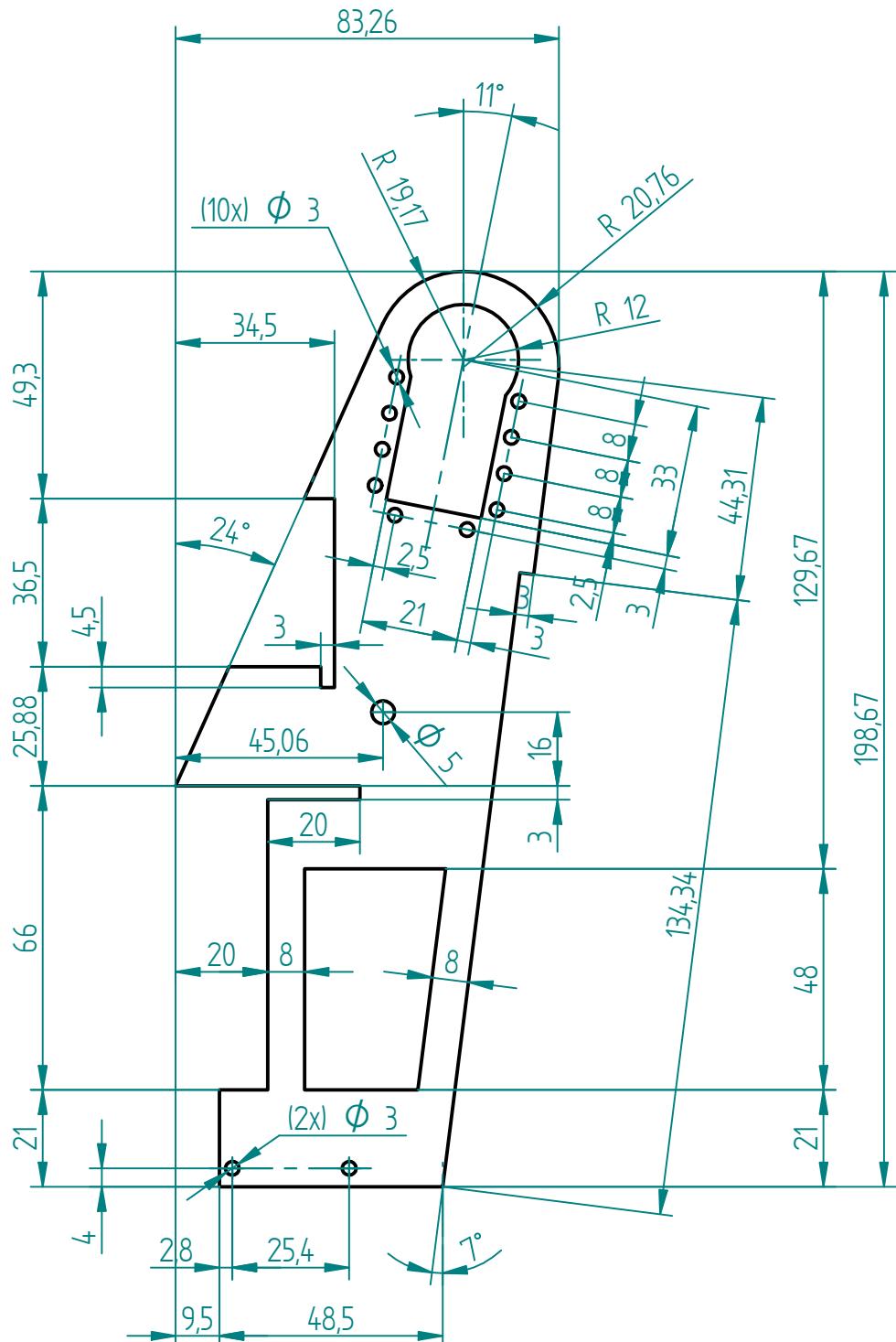
DIMENSIONES ACOTADAS EN MILIMETROS		HOJA	NUMERO PLANO	REV
TOLERANCIA GENERAL ±0.2		A4	08-00-02-022	0
ANGULO ±1°		ESCALA: 1:1,5	PESO:	HOJA 1 DE 1





DIBUJADO	M.I.G	17-02-15	Proyecto Eurobot 2015		
MATERIAL	TRESPA				
ESPESOR	3 MM				
CANTIDAD	1 UNIDADES		TITULO		
DIMENSIONES ACOTADAS EN MILIMETROS			SOporte DYNAMIXEL FIJO LATERAL DER.		
TOLERANCIA GENERAL ±0.2			HOJA	NUMERO PLANO	REV
ANGULO ±1°			A4	08-00-03-023	0
			ESCALA:	1:1,5	PESO:
			HOJA 1 DE 1		



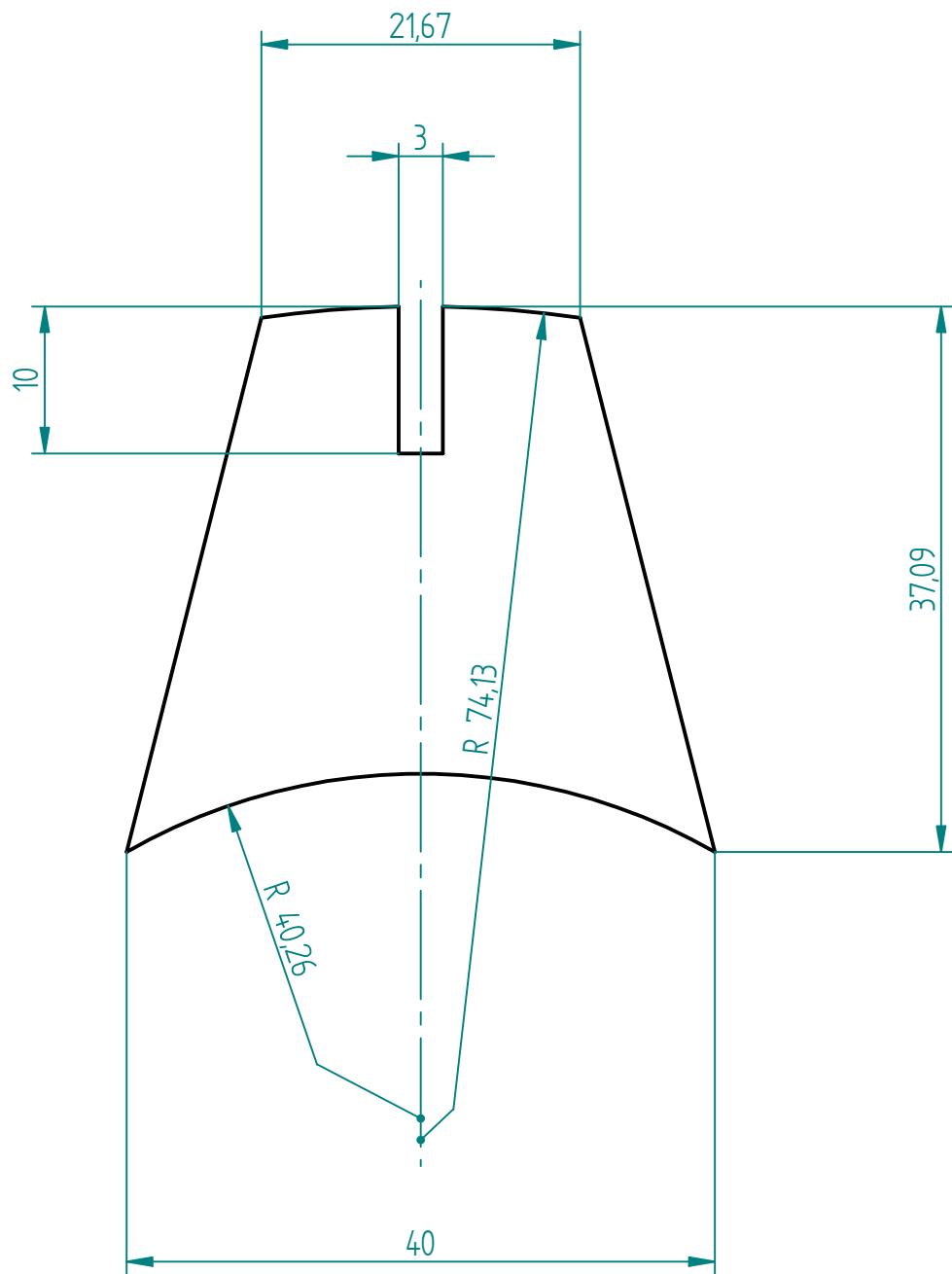


	NOMBRE	FECHA
DIBUJADO	M.I.G	16-02-15
MATERIAL	TRESPA	
ESPESOR	3 MM	
CANTIDAD	1 UNIDADES	
DIMENSIONES ACOTADAS EN MILIMETROS		
TOLERANCIA GENERAL $\pm 0.2$		
ANGULO $\pm 1^\circ$		

Proyecto Eurobot 2015

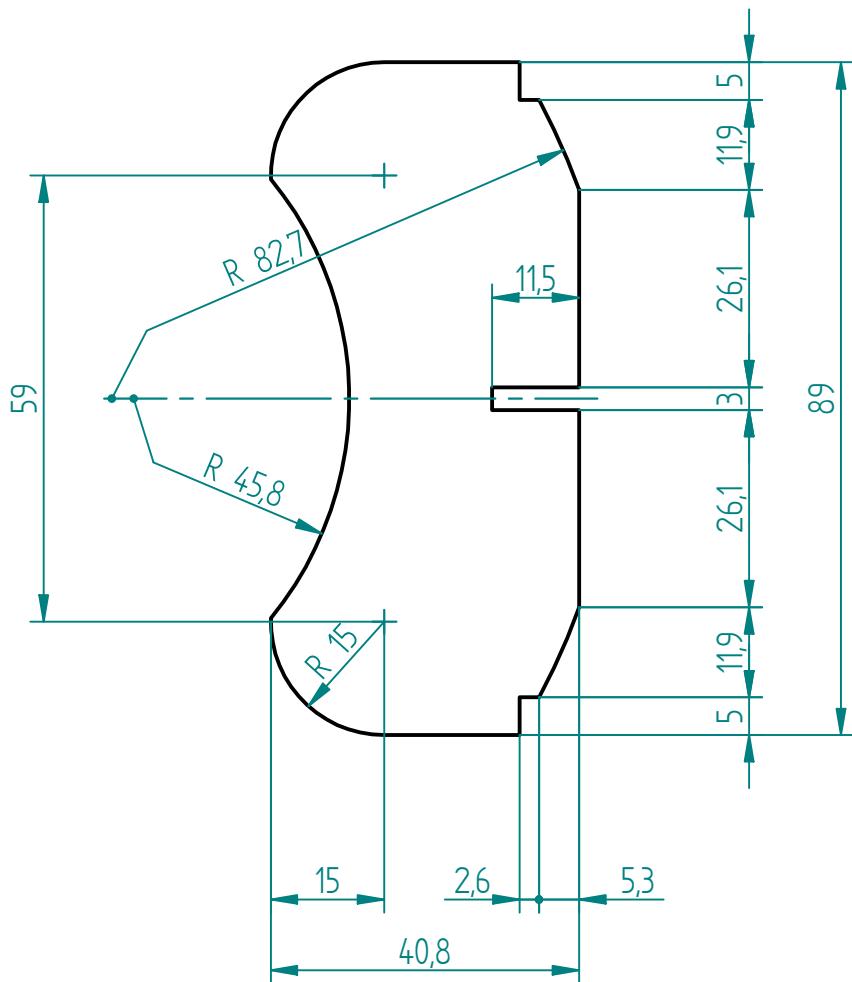
TITULO	SOporte DYNAMIXEL FIJO	
HOJA	NUMERO PLANO	REV
A4	08-00-04-024	0
ESCALA: 1:1,5	PESO:	HOJA 1 DE 1





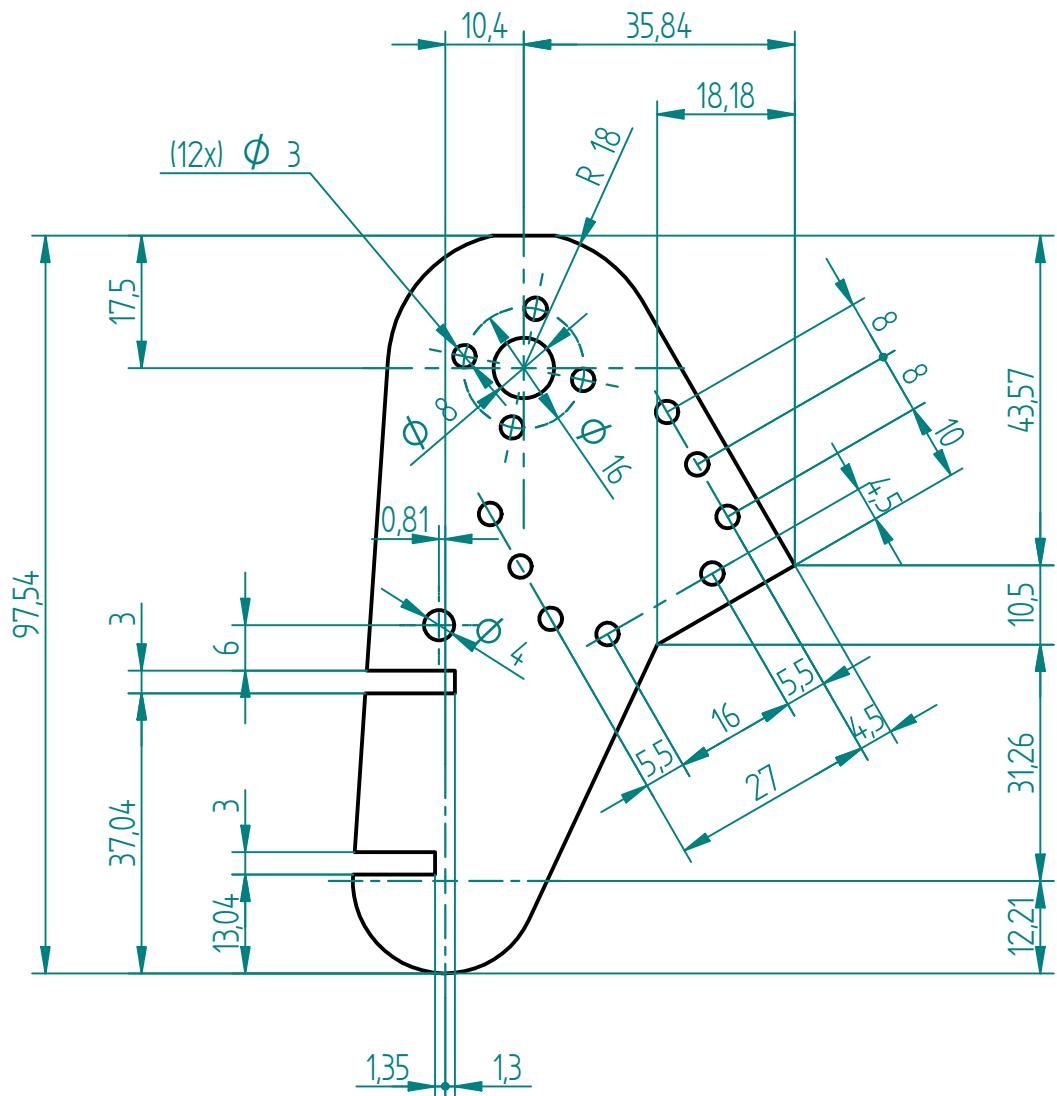
NOMBRE	M.I.G	FECHA	Proyecto Eurobot 2015		
DIBUJADO		16-02-15			
MATERIAL	TRESPA		TITULO		
ESPESOR	3 MM		APOYO INFERIOR		
CANTIDAD	1 UNIDADES				
DIMENSIONES ACOTADAS EN MILIMETROS TOLERANCIA GENERAL $\pm 0.2$ ANGULO $\pm 1^\circ$			HOJA	NUMERO PLANO	REV
			A4	08-01-01-041	0
			ESCALA:	PESO:	HOJA 1 DE 1





NOMBRE	M.I.G	FECHA	Proyecto Eurobot 2015		
DIBUJADO		16-02-15			
MATERIAL	TRESPA		TITULO		
ESPESOR	3 MM		APOYO SUPERIOR		
CANTIDAD	1 UNIDADES				
DIMENSIONES ACOTADAS EN MILIMETROS TOLERANCIA GENERAL $\pm 0.2$ ANGULO $\pm 1^\circ$			HOJA	NUMERO PLANO	REV
			A4	08-01-02-042	0
			ESCALA:	PESO:	HOJA 1 DE 1



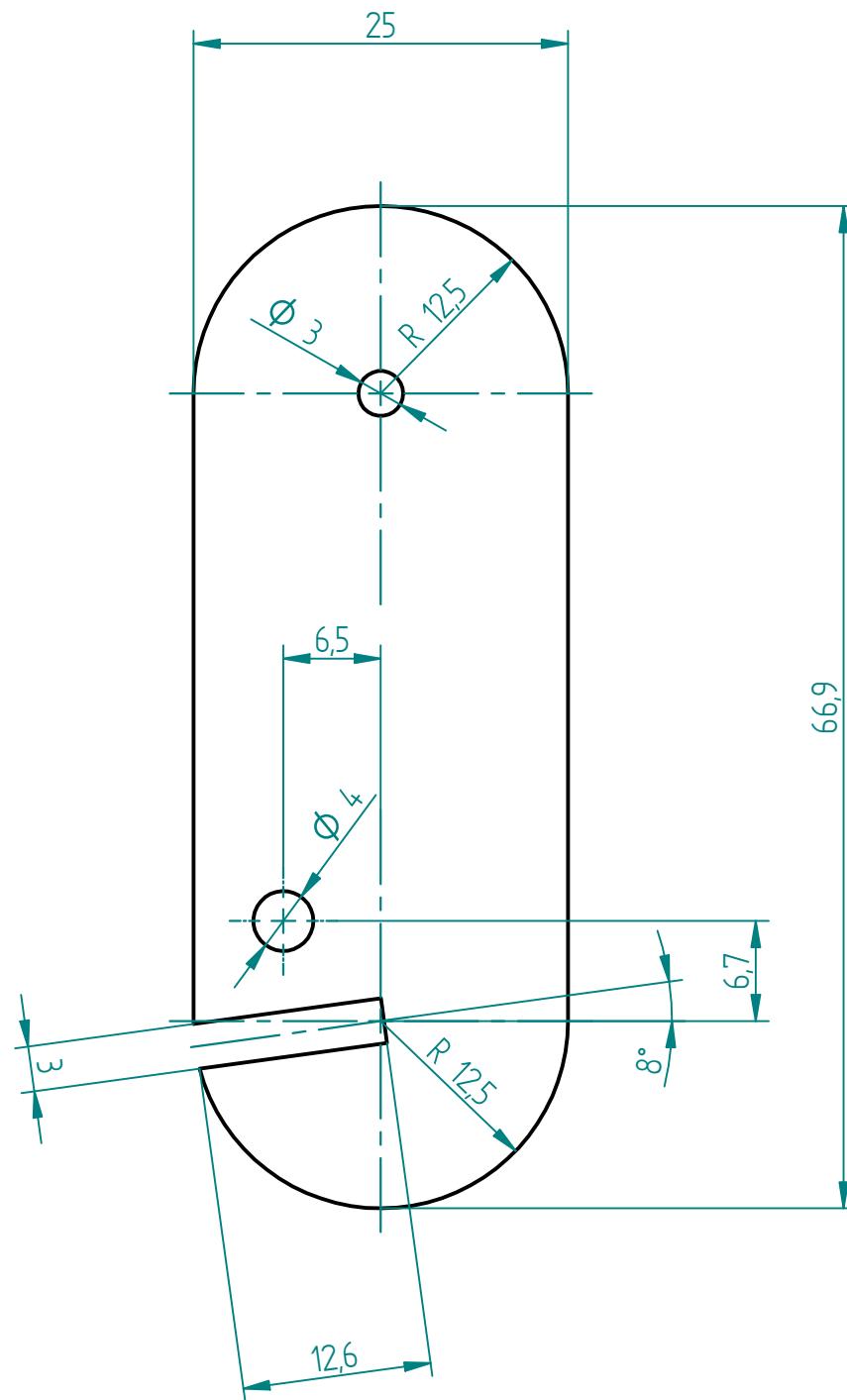


NOMBRE	FECHA
DIBUJADO	M.I.G 16-02-15
MATERIAL	TRESPA
ESPESOR	3 MM
CANTIDAD	1 UNIDADES
DIMENSIONES ACOTADAS EN MILIMETROS	
TOLERANCIA GENERAL ±0,2	
ANGULO ±1°	

## Proyecto Eurobot 2015

TITULO		
BRAZO SOPORTE DYNAMIXEL MOVIL		
HOJA	NUMERO PLANO	REV
A4	08-01-03-043	0
ESCALA:	PESO:	HOJA 1 DE 1





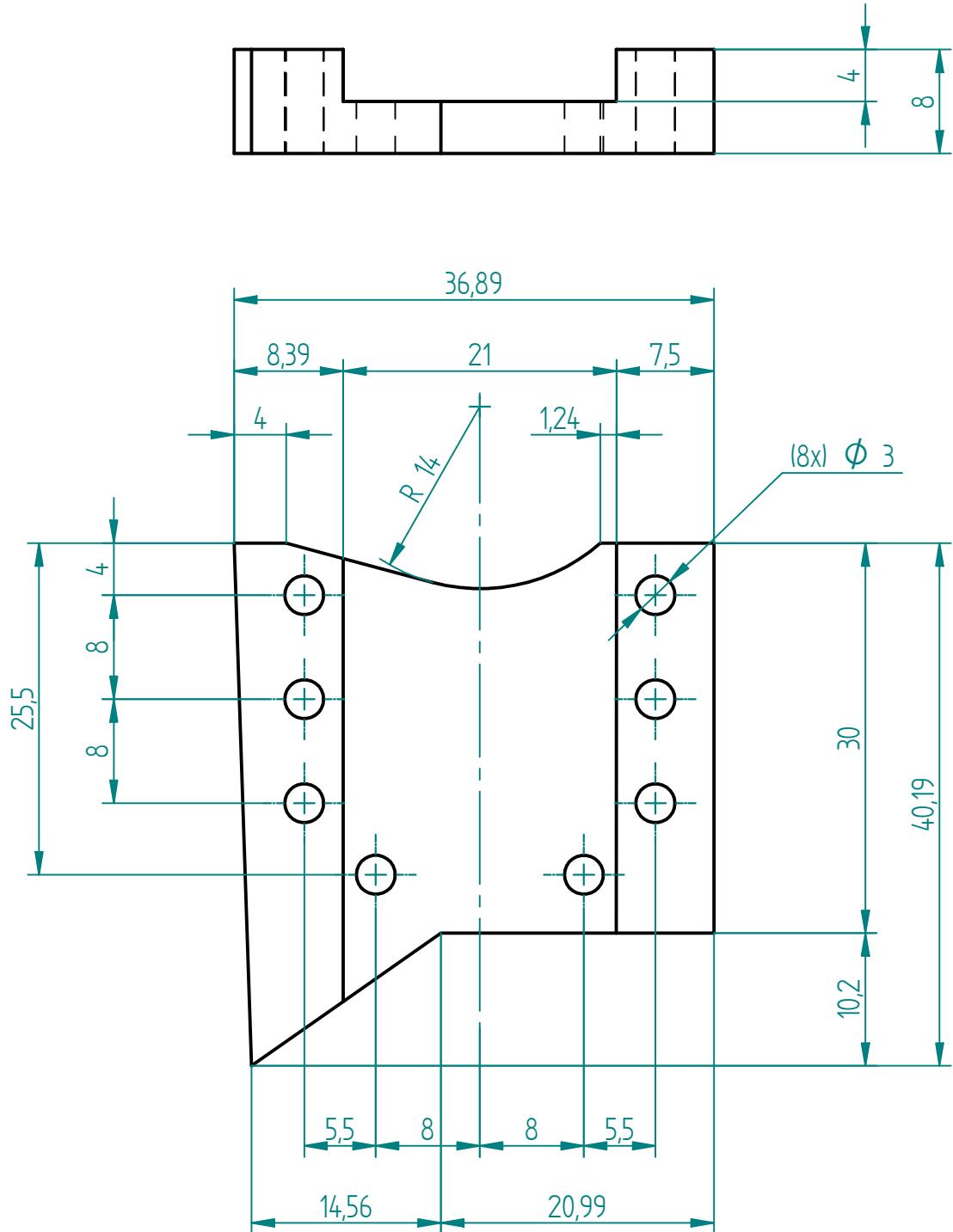
	NOMBRE	FECHA
DIBUJADO	M.I.G	16-02-15
MATERIAL	TRESPA	
ESPESOR	3 MM	
CANTIDAD	1 UNIDADES	
DIMENSIONES ACOTADAS EN MILIMETROS		
TOLERANCIA GENERAL $\pm 0.2$		
ANGULO $\pm 1^\circ$		

## Proyecto Eurobot 2015

TITULO BRAZO ESTABILIZADOR SOPORTE  
DYNAMIXEL

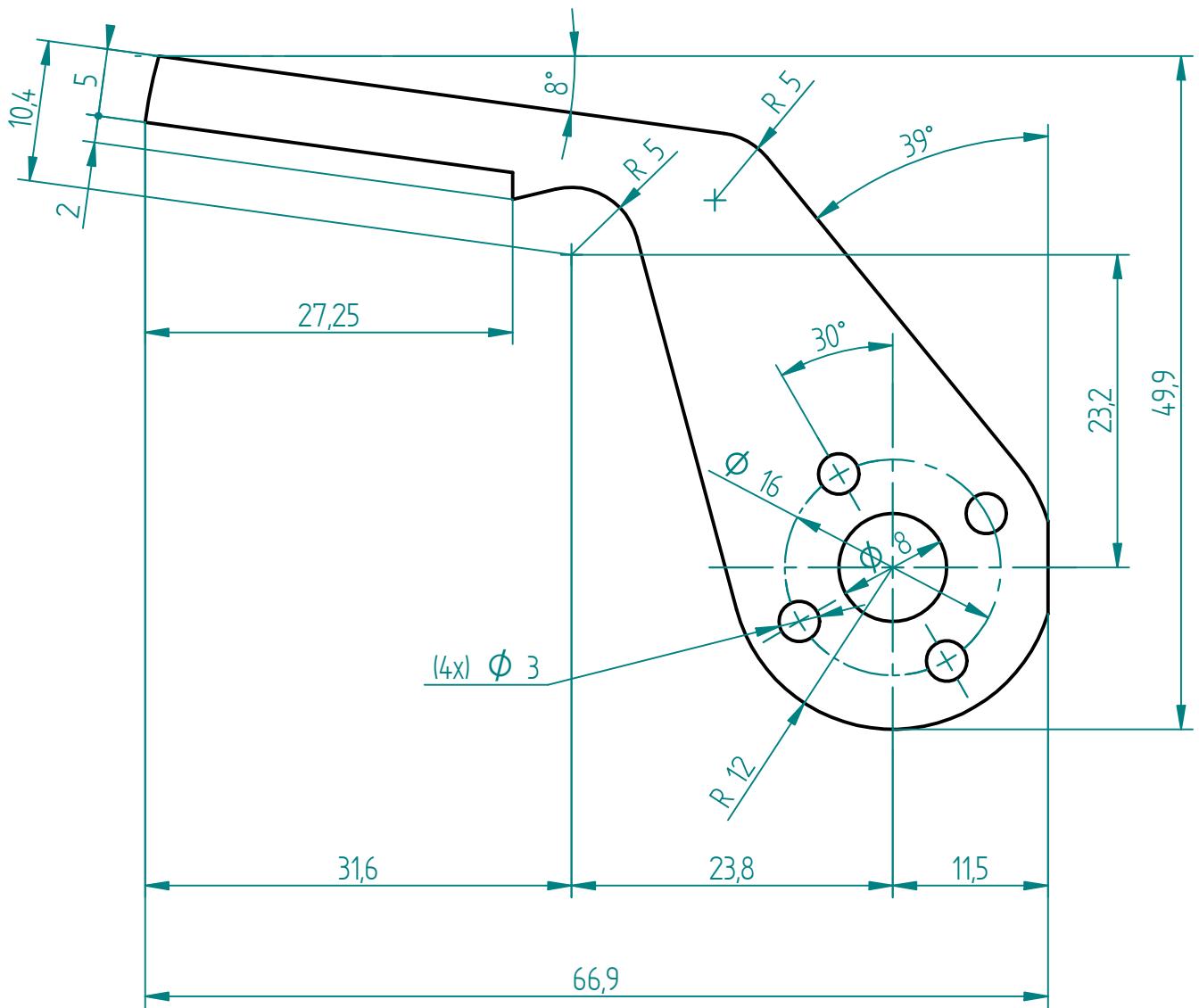
HOJA A4	NUMERO PLANO 08-01-04-044	REV 0
ESCALA: 2:1	PESO:	HOJA 1 DE 1





NOMBRE	M.I.G	FECHA	Proyecto Eurobot 2015		
DIBUJADO		16-02-15			
MATERIAL	TRESPA		TITULO		
ESPESOR	8 MM		SEPARADOR DYNAMIXEL MOVIL		
CANTIDAD	1 UNIDADES				
DIMENSIONES ACOTADAS EN MILIMETROS			HOJA	NUMERO PLANO	REV
TOLERANCIA GENERAL ±0.2			A4	08-01-05-045	0
ANGULO ±1°			ESCALA:	PESO:	HOJA 1 DE 1



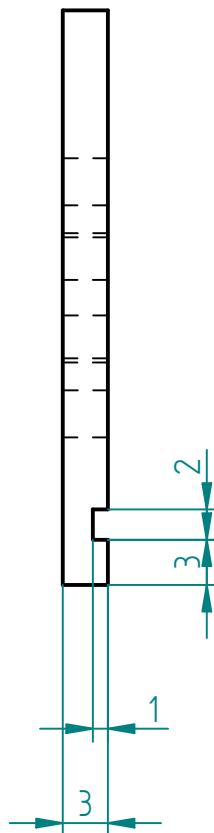
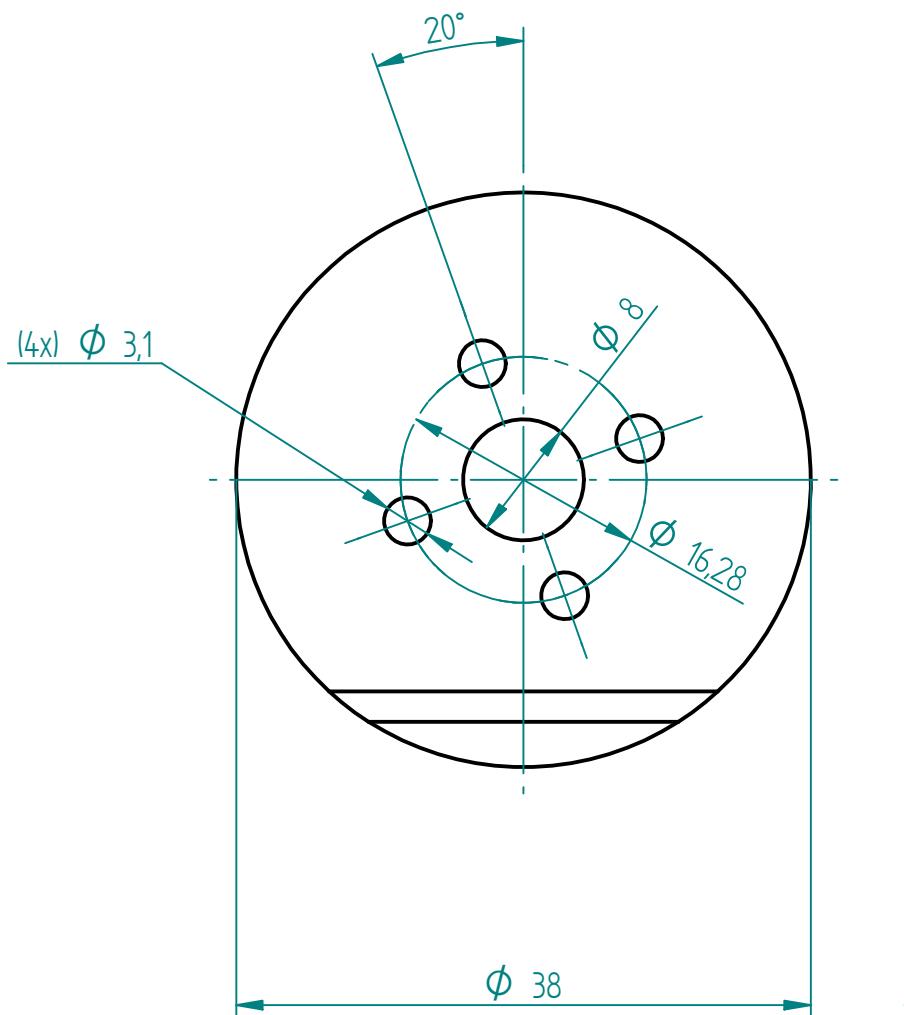


	NOMBRE	FECHA
DIBUJADO	M.I.G	16-02-15
MATERIAL	TRESPA	
ESPESOR	3 MM	
CANTIDAD	1 UNIDADES	
DIMENSIONES ACOTADAS EN MILIMETROS		HOJA
TOLERANCIA GENERAL $\pm 0.2$		NUMERO PLANO
ANGULO $\pm 1^\circ$		A4 08-01-06-046

Proyecto Eurobot 2015

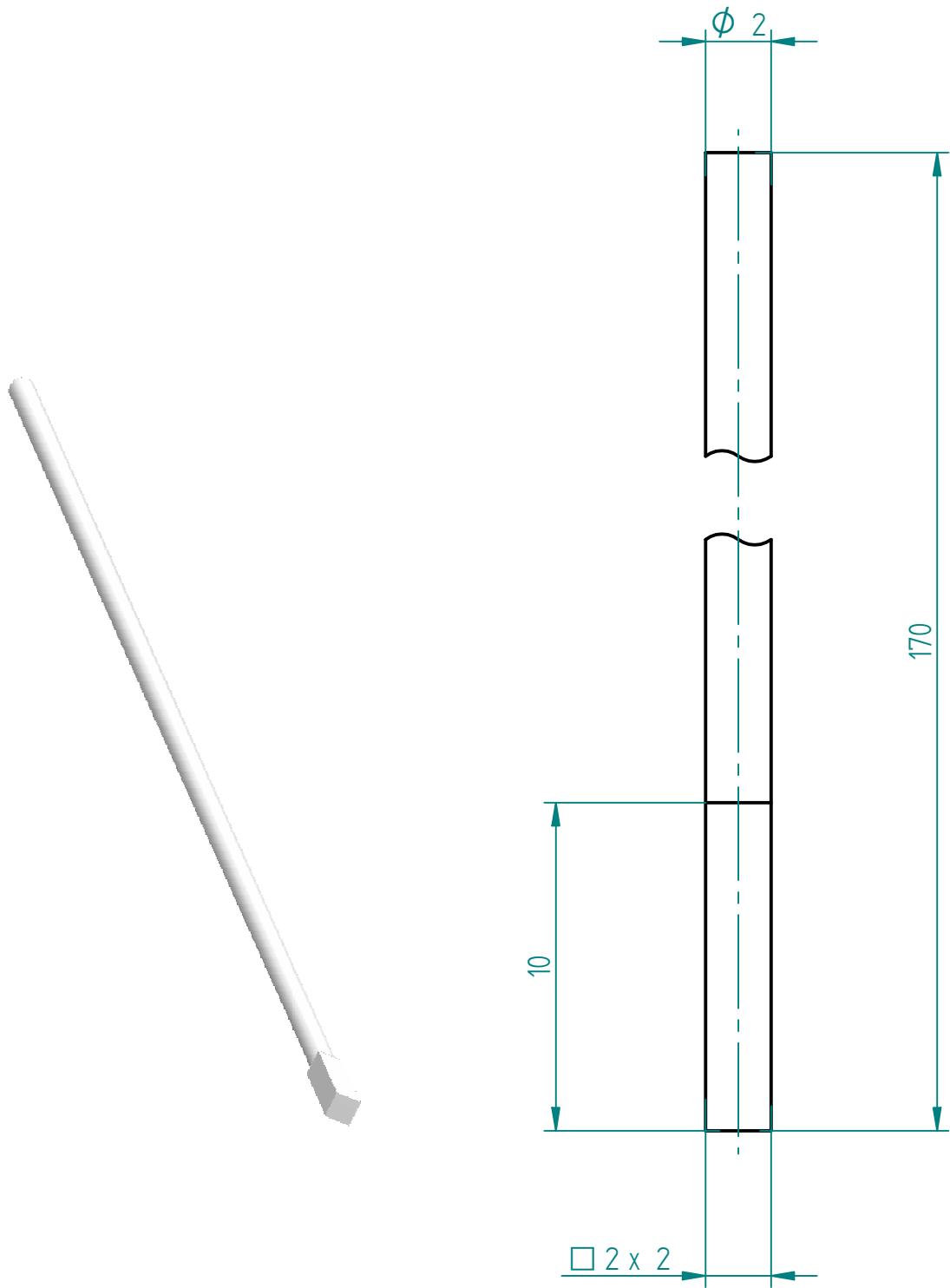
TITULO	PINZA	REV 0
ESCALA: 2:1	PESO:	HOJA 1 DE 1





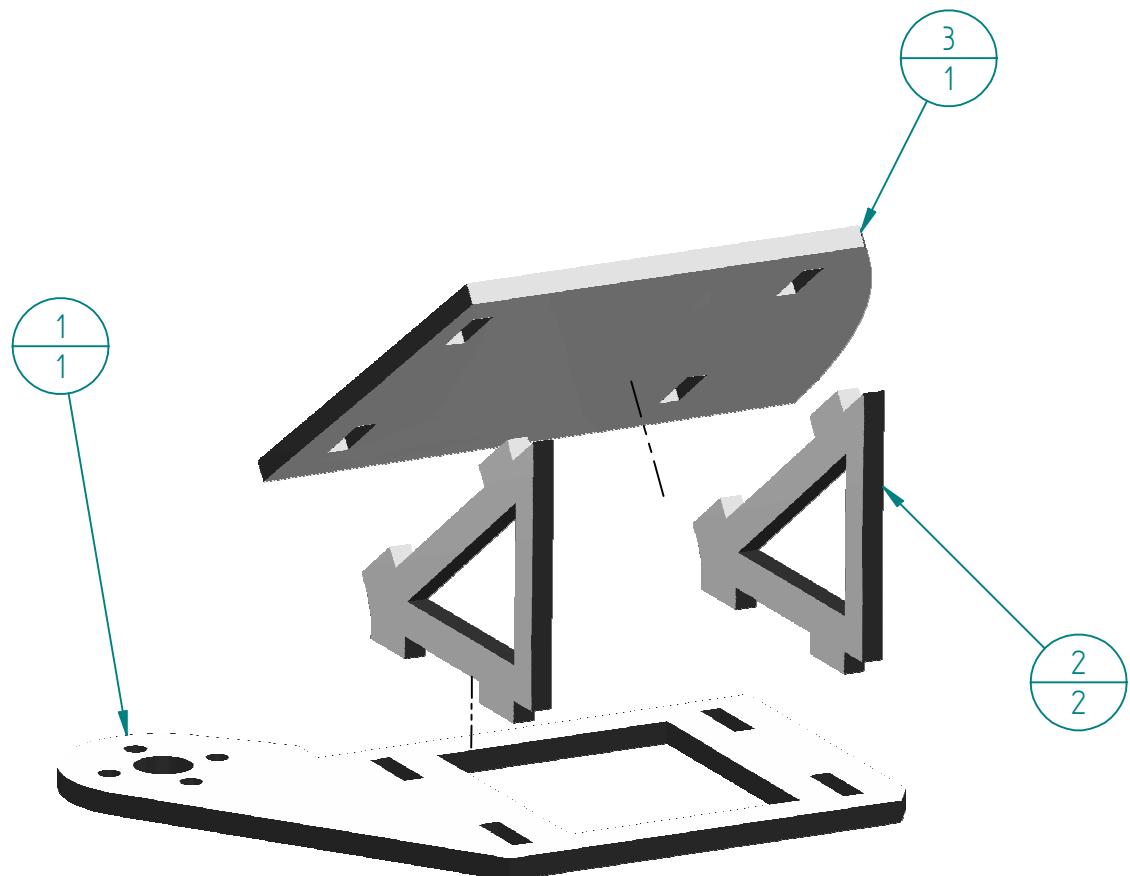
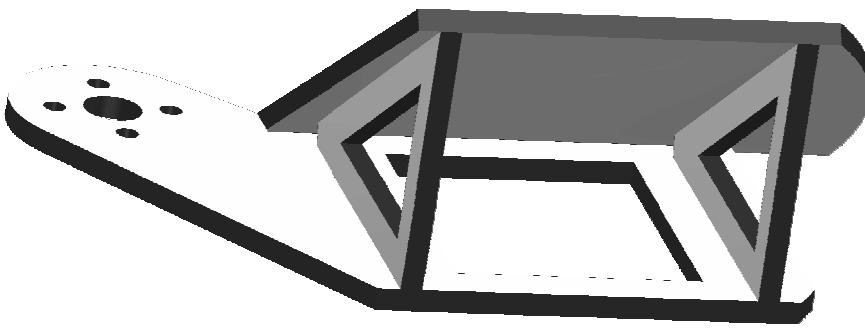
NOMBRE	M.I.G	FECHA	Proyecto Eurobot 2015		
DIBUJADO					
MATERIAL	TRESPA		TITULO		
ESPESOR	3 MM		DISCO BRAZO CLAQUETA		
CANTIDAD	2 S/D + 2 M/C				
DIMENSIONES ACOTADAS EN MILIMETROS			HOJA	NUMERO PLANO	REV
TOLERANCIA GENERAL ±0.2			A4	09-00-01-071	0
ANGULO ±1°			ESCALA:	PESO:	HOJA 1 DE 1





NOMBRE	FECHA	Proyecto Eurobot 2015		
DIBUJADO	M.I.G	XX-XX-XX		
MATERIAL	FIBRA DE VIDRIO	TITULO		
ESPESOR	Ø 2 MM	VARILLA BRAZO CLAQUETA		
CANTIDAD	2 UNIDADES			
DIMENSIONES ACOTADAS EN MILIMETROS		HOJA	NUMERO PLANO	REV
TOLERANCIA GENERAL ±0.2		A4	09-00-02-000	0
ANGULO ±1°		ESCALA:	5.1	PESO:
				HOJA 1 DE 1

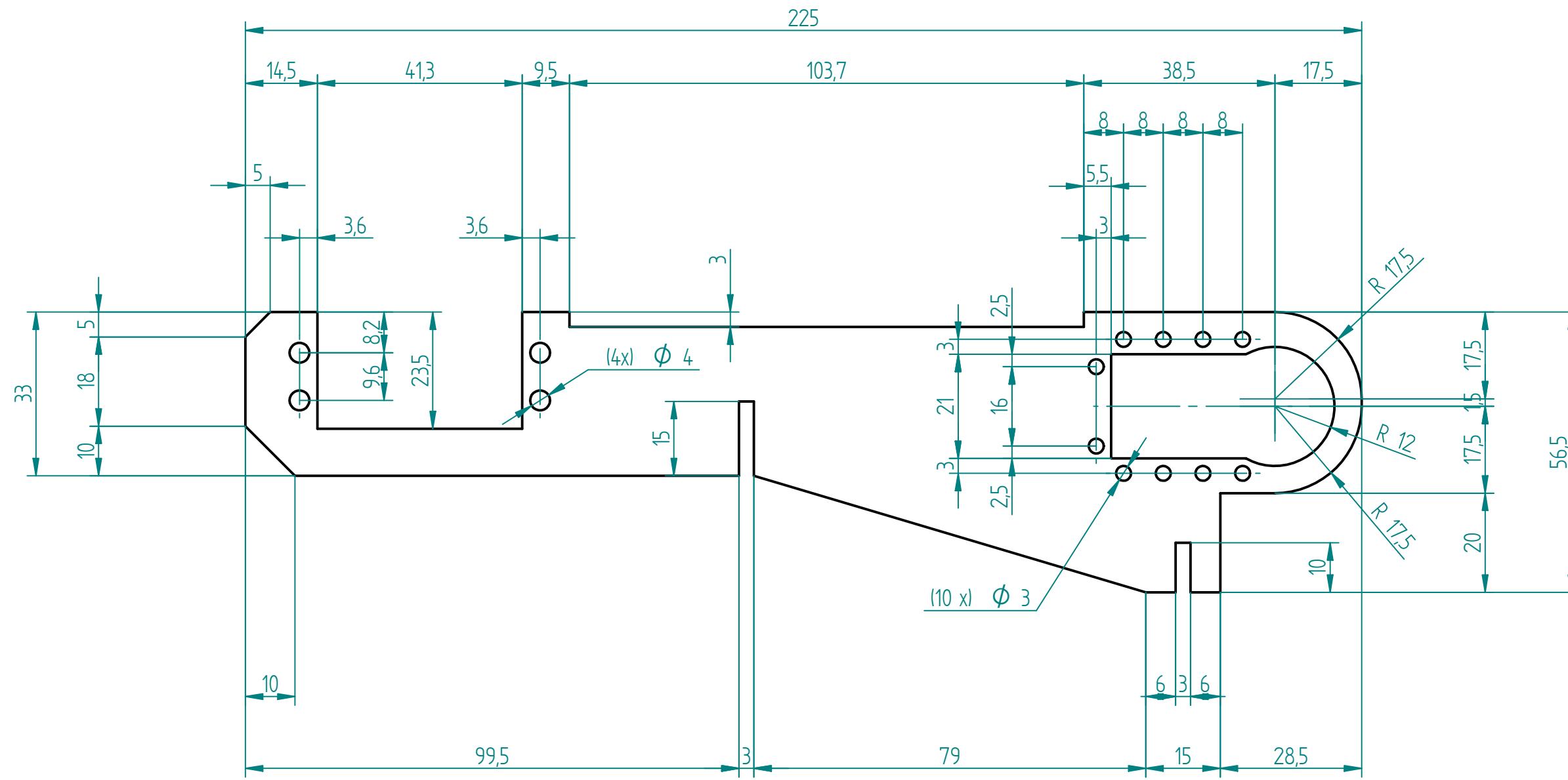




ITEM	PLANO	N	MATERIAL	MASA
1	Base Brazos Palomitas	1	Epoxy, cast	0,012 kg
2	Pendiente Brazos	2	Epoxy, cast	0,003 kg
3	Tapa Brazos Palomitas	1	Epoxy, cast	0,018 kg

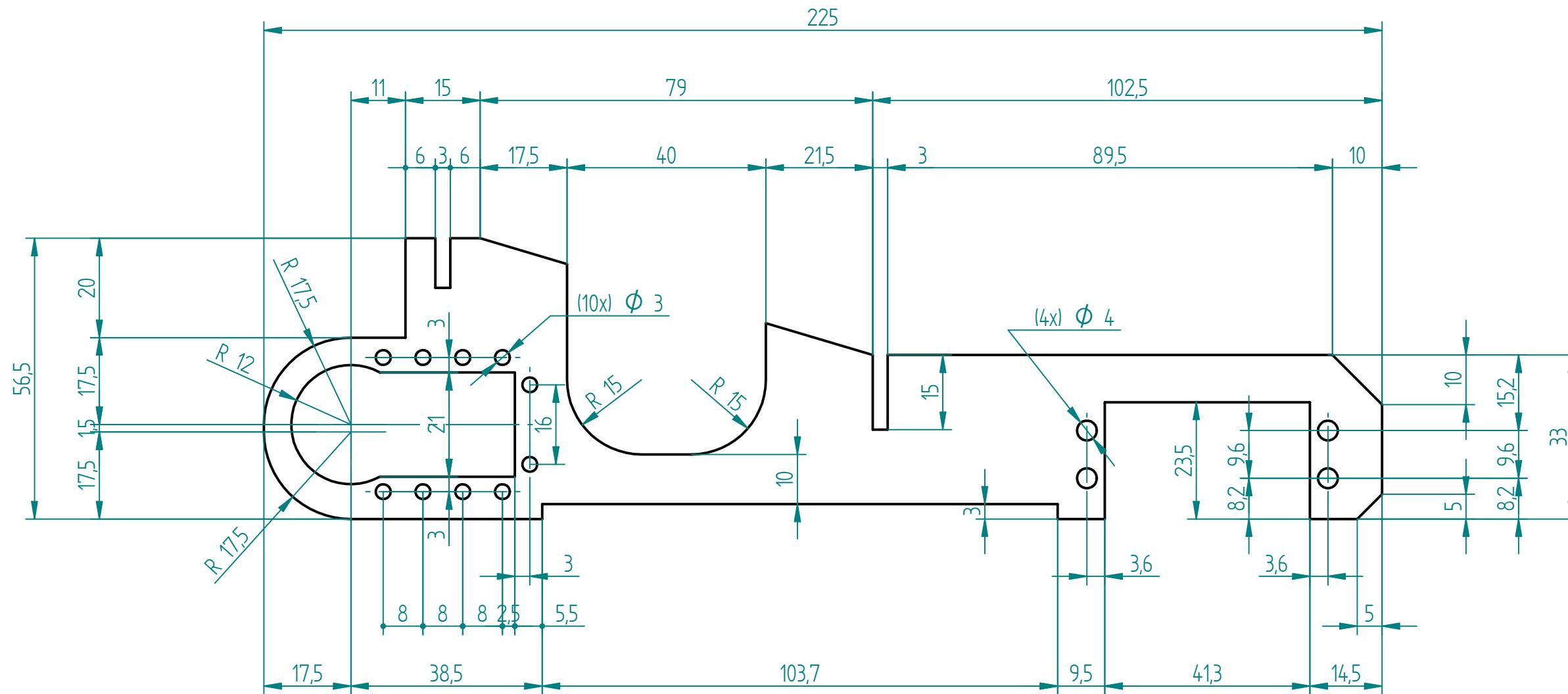
DIBUJADO	NOMBRE	FECHA	Proyecto Eurobot 2015		
	M.I.G	24-01-15			
MATERIAL					
ESPESOR					
CANTIDAD	1 S/D + 1 M/C				
DIMENSIONES ACOTADAS EN MILIMETROS TOLERANCIA GENERAL ±0.2 ANGULO ±1°			HOJA	NUMERO PLANO	REV
			A4	10-01-00-E01	0
			ESCALA:	PESO:	HOJA 1 DE 1





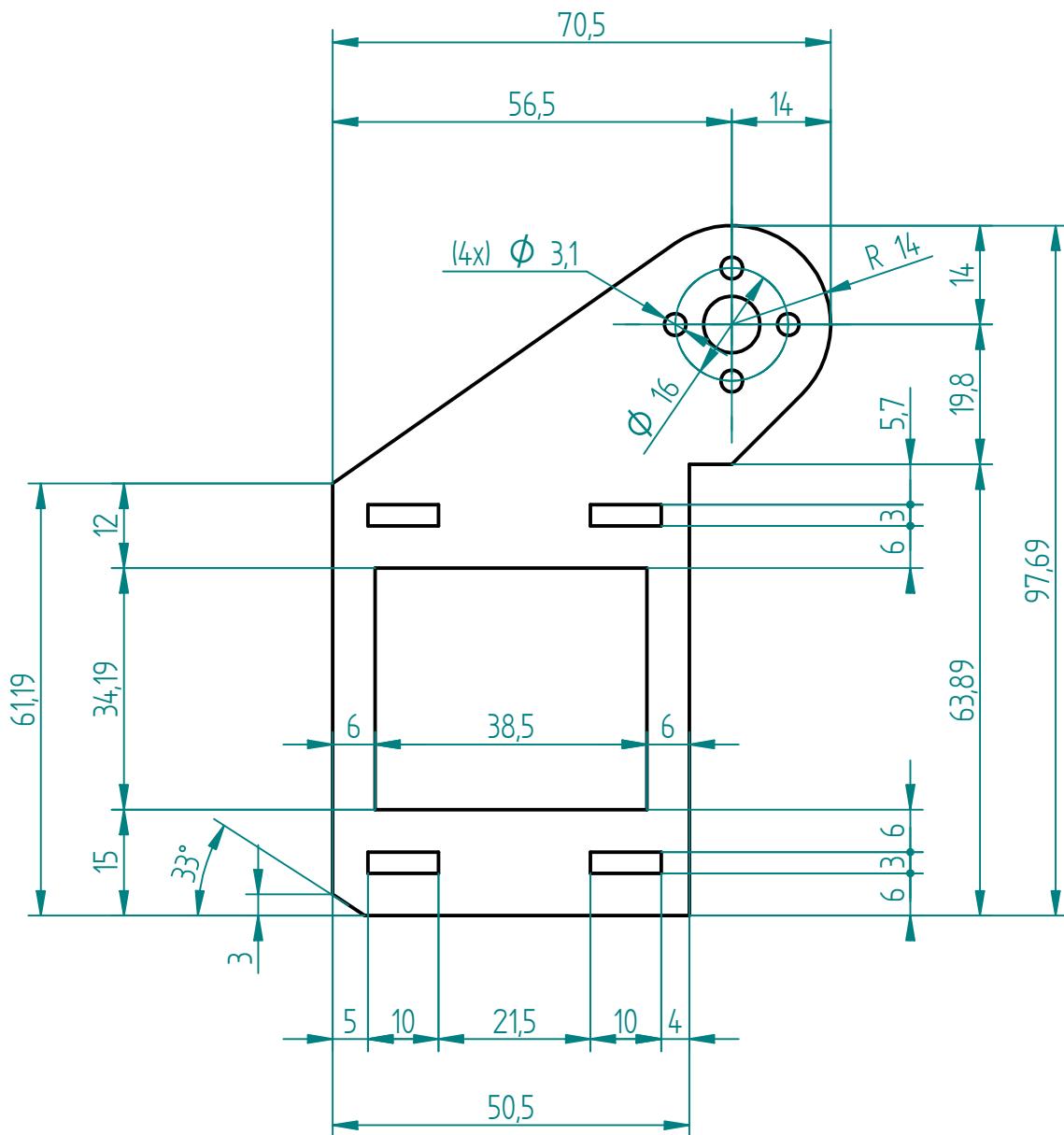
	NOMBRE	FECHA	Proyecto Eurobot 2015		
DIBUJADO	M.I.G	07-01-15			
MATERIAL	TRESPA		TITULO	ESTRUCTURA SOPORTE SERVO	
ESPESOR	3 MM			BRAZO PALOMITAS	
CANTIDAD	1 UNIDADES		HOJA A3	NUMERO PLANO 10-01-01-037	REV 0
DIMENSIONES ACOTADAS EN MILIMETROS					
TOLERANCIA GENERAL ±0.2					
ANGULO ±1°					
ESCALA: 1:1		PESO:	HOJA 1 DE 1		





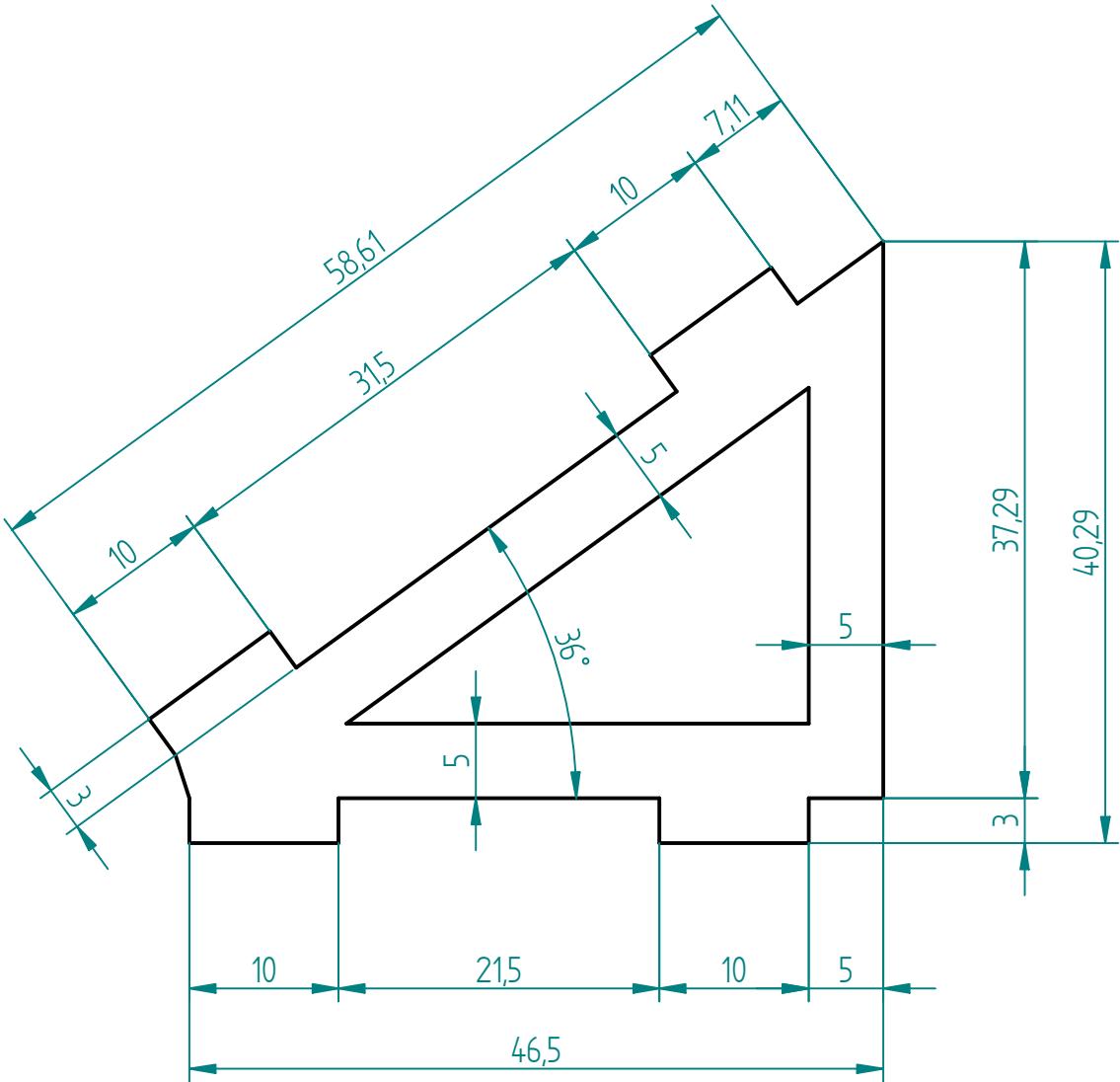
	NOMBRE	FECHA	Proyecto Eurobot 2015		
DIBUJADO	M.I.G	21-02-15			
MATERIAL	TRESPA		TITULO	SOporte SERVO BRAZOS	
ESPESOR	3 MM			PALOMITAS DERECHO	
CANTIDAD	1 UNIDADES				
DIMENSIONES ACOTADAS EN MILIMETROS TOLERANCIA GENERAL ±0.2			HOJA A3	NUMERO PLANO 10-01-02-039	REV 0
ANGULO ±1°			ESCALA: 1:1	PESO:	HOJA 1 DE 1





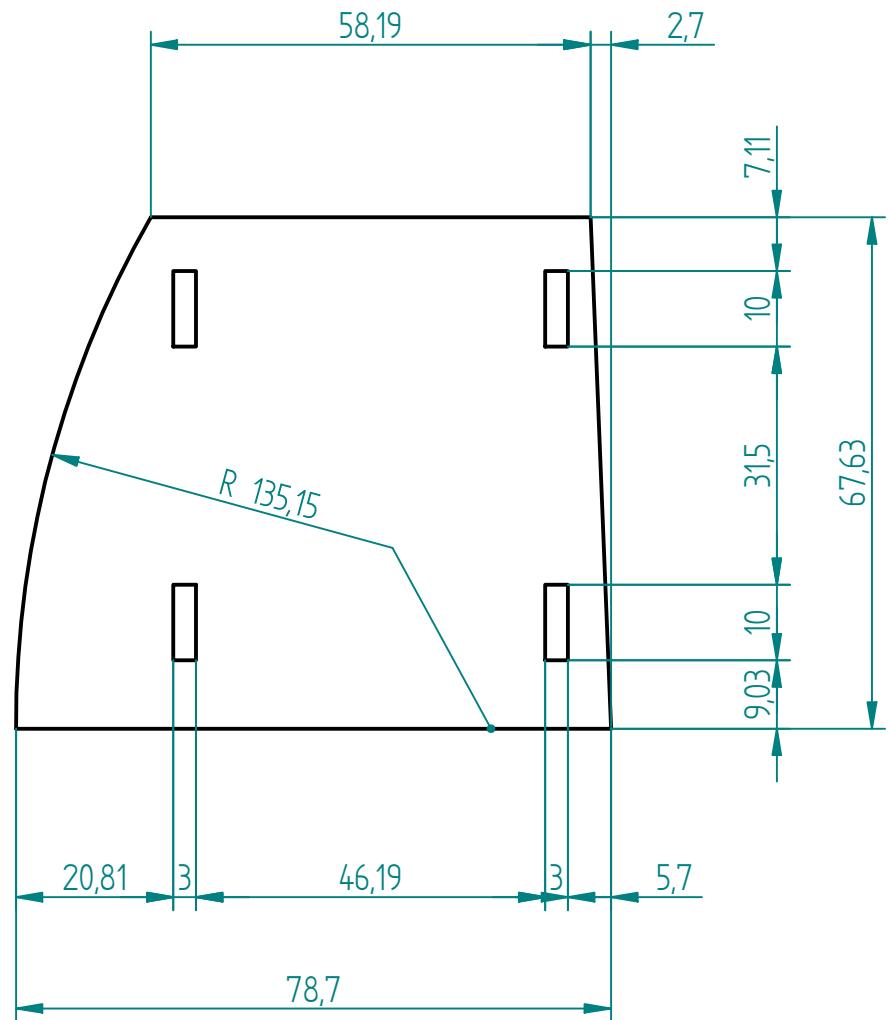
NOMBRE	M.I.G	FECHA	Proyecto Eurobot 2015		
DIBUJADO					
MATERIAL	TRESPA				
ESPESOR	3 MM				
CANTIDAD	2 UNIDADES		TITULO		
DIMENSIONES ACOTADAS EN MILIMETROS TOLERANCIA GENERAL $\pm 0.2$ ANGULO $\pm 1^\circ$			HOJA	NUMERO PLANO	REV
			A4	10-01-04-083	0
			ESCALA: 1:1	PESO:	HOJA 1 DE 1





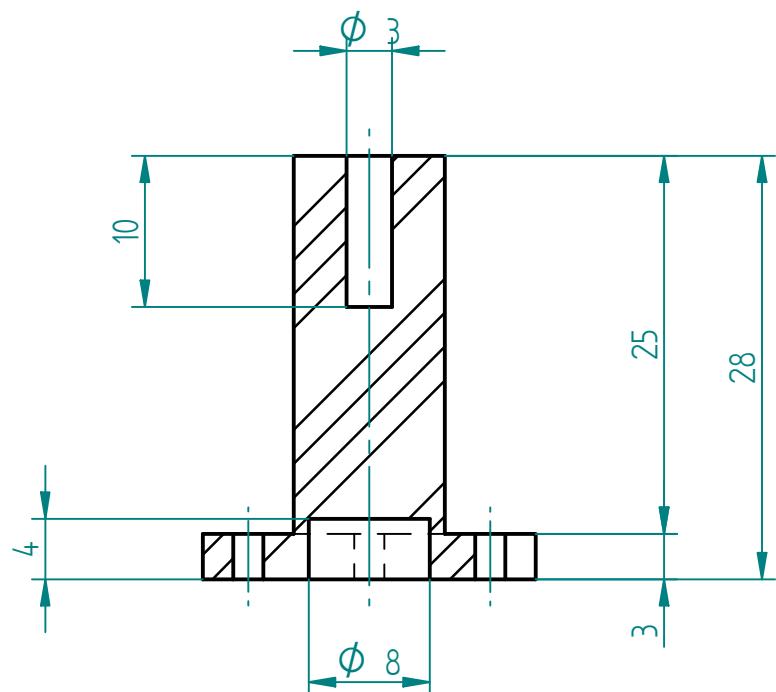
NOMBRE	M.I.G	FECHA	Proyecto Eurobot 2015		
DIBUJADO					
MATERIAL	TRESPA		TITULO		
ESPESOR	3 MM		PENDIENTE BRAZOS PALOMITAS		
CANTIDAD	4 UNIDADES				
DIMENSIONES ACOTADAS EN MILIMETROS TOLERANCIA GENERAL $\pm 0.2$ ANGULO $\pm 1^\circ$			HOJA	NUMERO PLANO	REV
			A4	10-01-04-082	0
			ESCALA: 2:1	PESO:	HOJA 1 DE 1



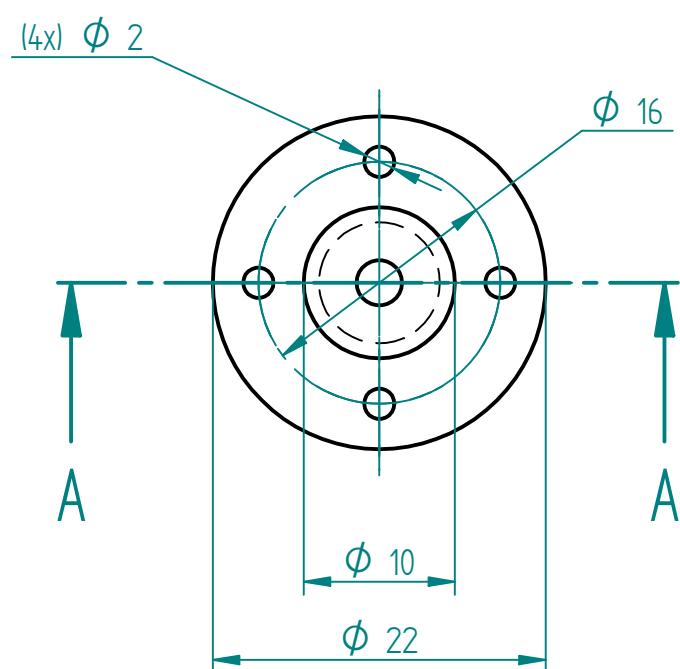


NOMBRE		FECHA	Proyecto Eurobot 2015		
DIBUJADO	M.I.G	07-02-15			
MATERIAL	TRESPA		TITULO		
ESPESOR	3 MM		TAPA BRAZOS PALOMITAS		
CANTIDAD	2 UNIDADES				
DIMENSIONES ACOTADAS EN MILIMETROS			HOJA	NUMERO PLANO	REV
TOLERANCIA GENERAL ±0.2			A4	10-01-05-081	0
ANGULO ±1°			ESCALA: 1:1	PESO:	HOJA 1 DE 1



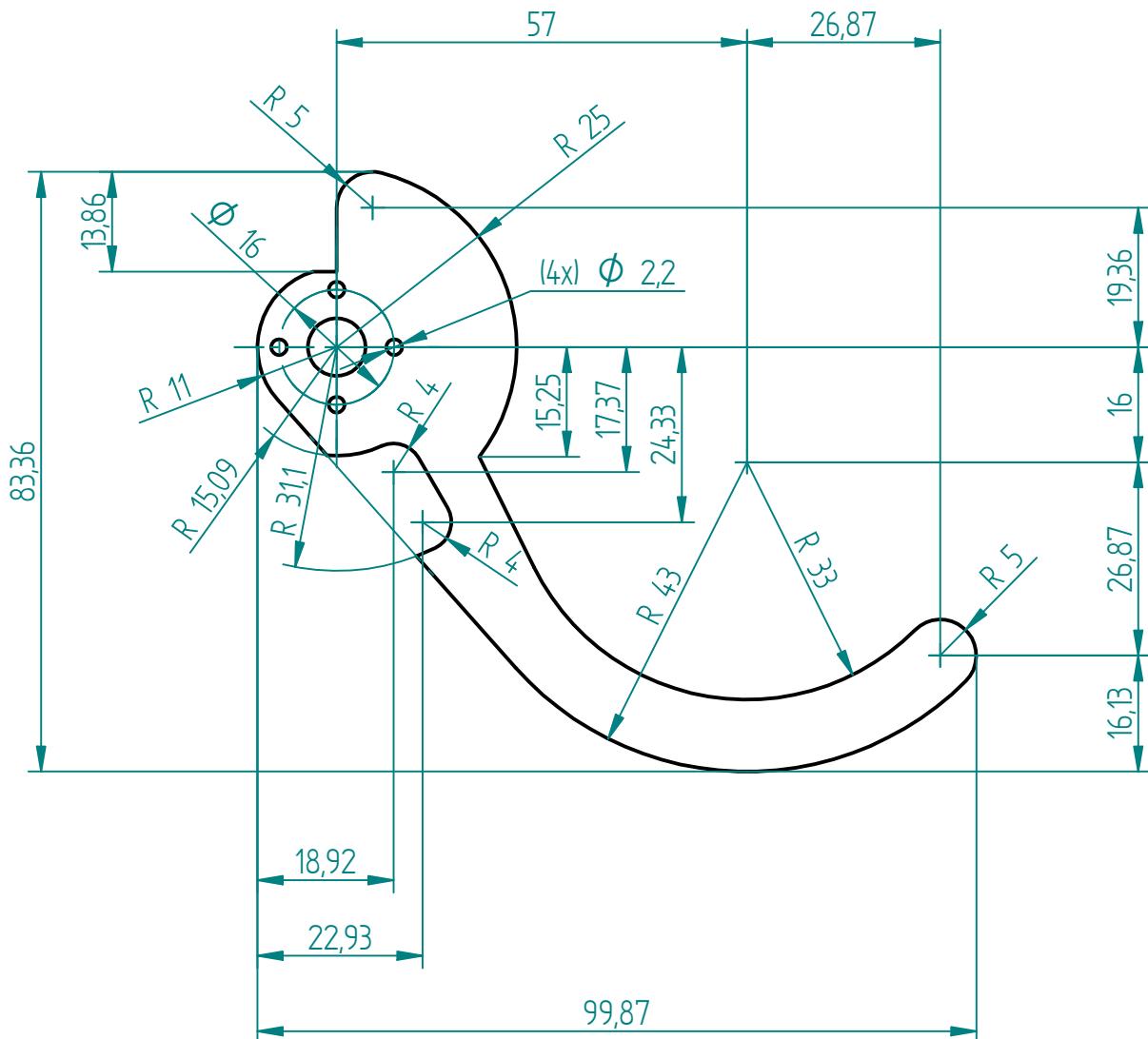


SECTION A-A



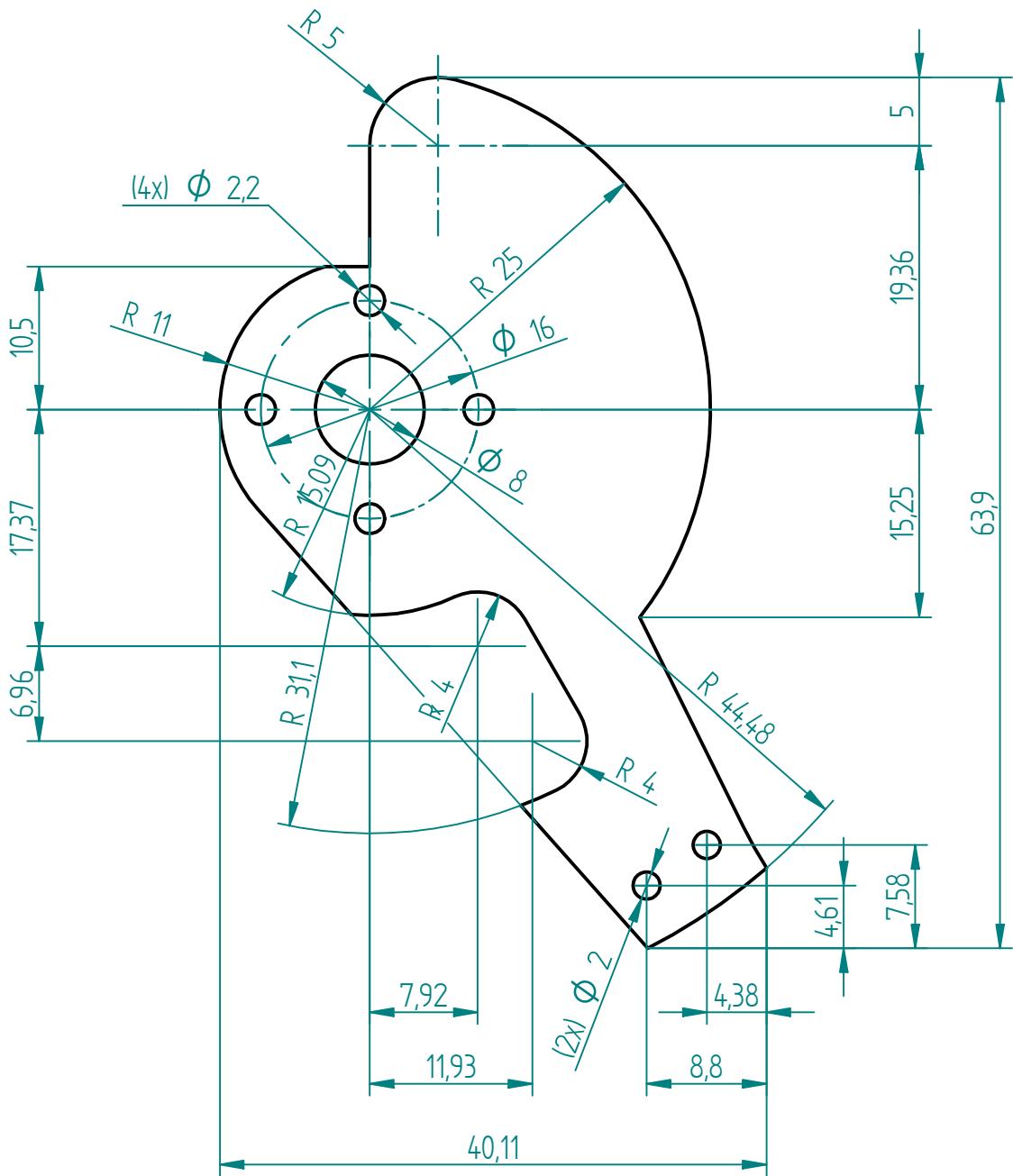
NOMBRE	M.I.G	FECHA	Proyecto Eurobot 2015		
DIBUJADO		24-01-15			
MATERIAL	ALUMINIO DURAL			TITULO	
ESPESOR				SUPLEMENTO SERVO BRAZO PALOMITAS	
CANTIDAD	2 UNIDADES				
DIMENSIONES ACOTADAS EN MILIMETROS			HOJA	NUMERO PLANO	REV
TOLERANCIA GENERAL $\pm 0.2$			A4	10-01-06-085	0
ANGULO $\pm 1^\circ$			ESCALA:	PESO:	HOJA 1 DE 1





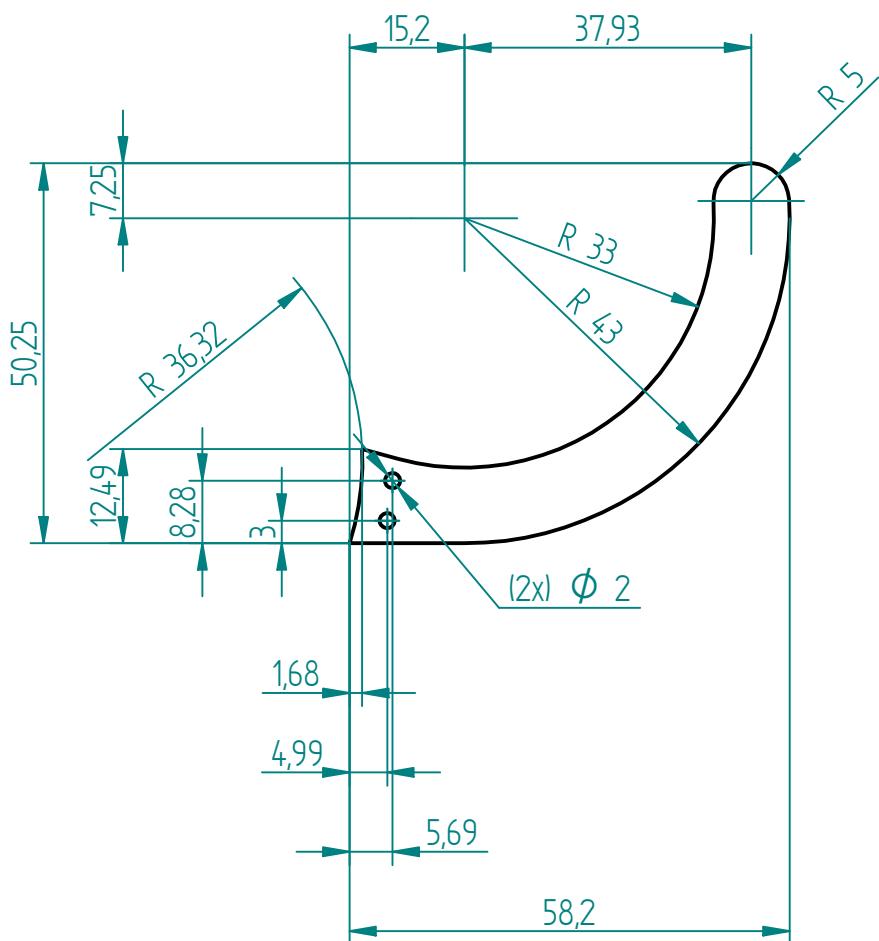
NOMBRE	M.I.G	FECHA	Proyecto Eurobot 2015		
DIBUJADO					
MATERIAL	TRESPA				
ESPESOR	3 MM				
CANTIDAD	1 UNIDADES		TITULO		
DIMENSIONES ACOTADAS EN MILIMETROS			PINZA COMPLETA		
TOLERANCIA GENERAL ±0.2			HOJA	NUMERO PLANO	REV
ANGULO ±1°			A4	11-00-01-086	0
			ESCALA:	PESO:	HOJA 1 DE 1





NOMBRE	M.I.G	FECHA	Proyecto Eurobot 2015		
DIBUJADO					
MATERIAL	TRESPA		TITULO		
ESPESOR	3 MM		PINZA PARTIDA EJE		
CANTIDAD	1 UNIDADES				
DIMENSIONES ACOTADAS EN MILIMETROS TOLERANCIA GENERAL ±0.2 ANGULO ±1°			HOJA	NUMERO PLANO	REV
			A4	11-00-02-087	0
			ESCALA:	PESO:	HOJA 1 DE 1





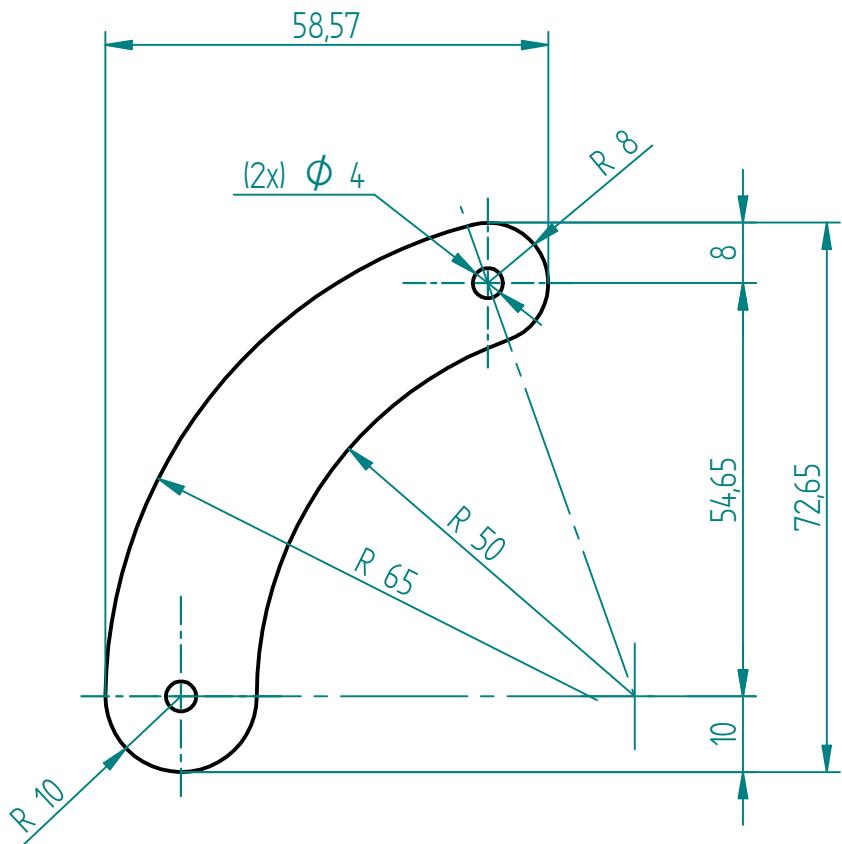
	NOMBRE	FECHA
DIBUJADO	M.I.G	08-02-15
MATERIAL	TRESPA	
ESPESOR	3 MM	
CANTIDAD	1 UNIDADES	
DIMENSIONES ACOTADAS EN MILIMETROS		
TOLERANCIA GENERAL $\pm 0.2$		
ANGULO $\pm 1^\circ$		

Proyecto Eurobot 2015

TITULO  
PINZA PARTIDA PUNTA

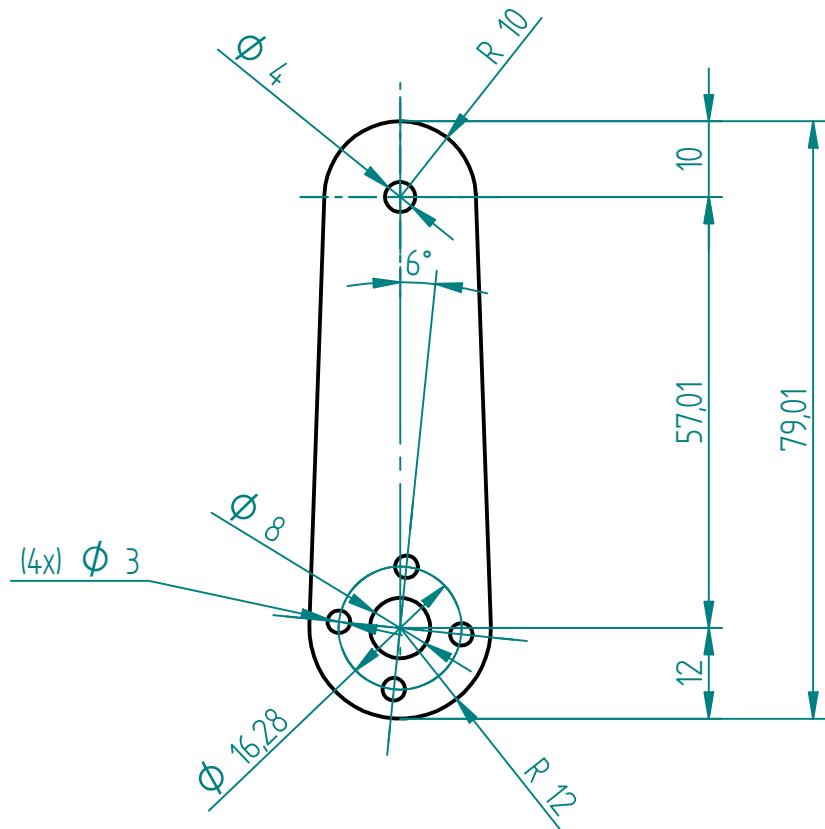
HOJA A4	NUMERO PLANO 11-00-03-088	REV 0
ESCALA: 2:1	PESO:	HOJA 1 DE 1





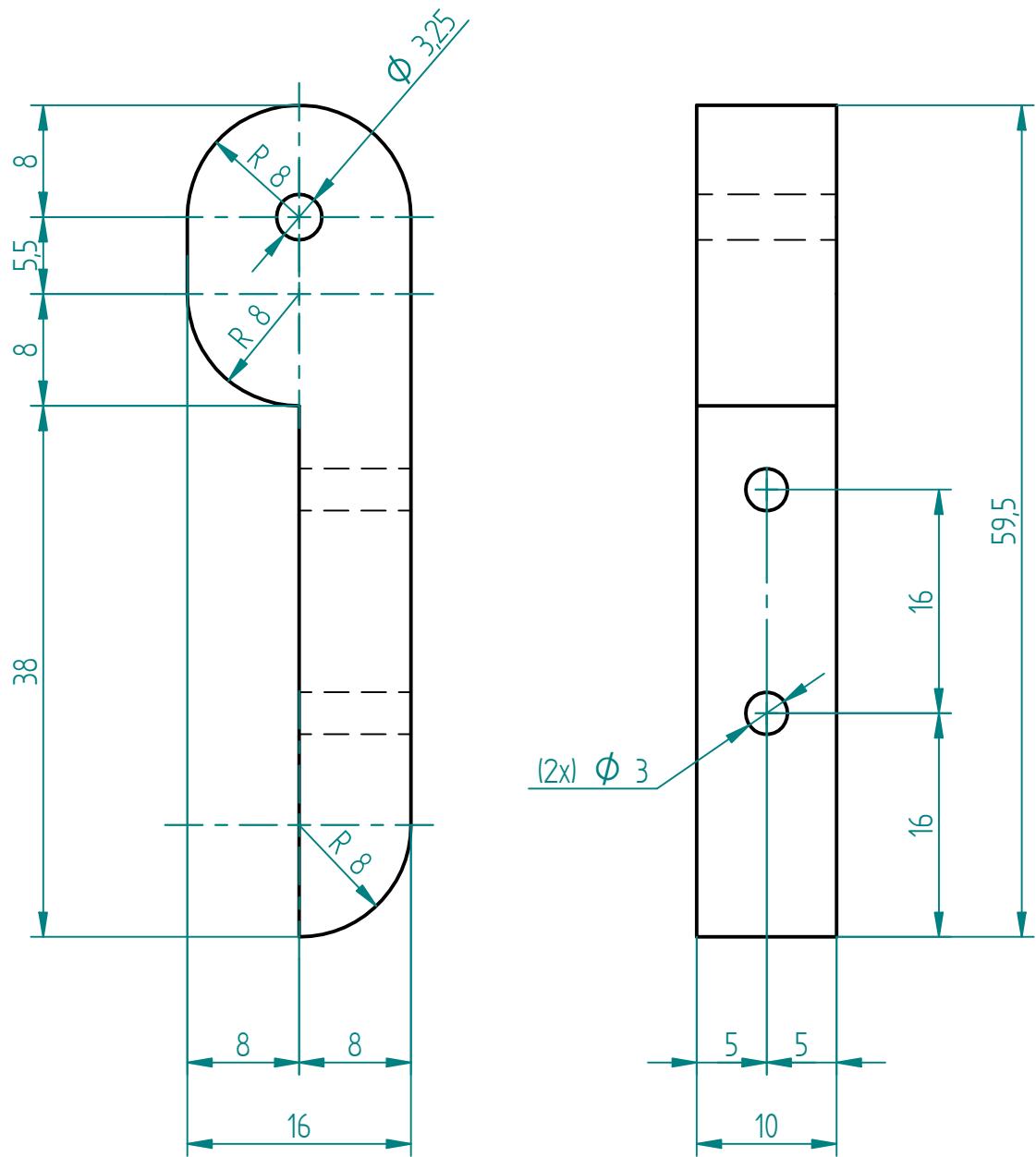
NOMBRE		FECHA	Proyecto Eurobot 2015		
DIBUJADO	M.I.G	12-02-15			
MATERIAL	TRESPA				TITULO
ESPESOR	3 MM				LEVA SERVO TRAMPILLA SUPERIOR
CANTIDAD	1 UNIDADES				
DIMENSIONES ACOTADAS EN MILIMETROS			HOJA	NUMERO PLANO	REV
TOLERANCIA GENERAL $\pm 0.2$			A4	12-00-01-075	0
ANGULO $\pm 1^\circ$			ESCALA: 1:1	PESO:	HOJA 1 DE 1





NOMBRE		FECHA	Proyecto Eurobot 2015		
DIBUJADO	M.I.G	12-02-15			
MATERIAL	TRESPA				
ESPESOR	3 MM				
CANTIDAD	1 UNIDADES	TITULO			BRAZO SERVO TRAMPILLA SUPERIOR
DIMENSIONES ACOTADAS EN MILIMETROS			HOJA	NUMERO PLANO	REV
TOLERANCIA GENERAL ±0.2			A4	12-00-02-076	0
ANGULO ±1°			ESCALA:	1:1	PESO:
					HOJA 1 DE 1



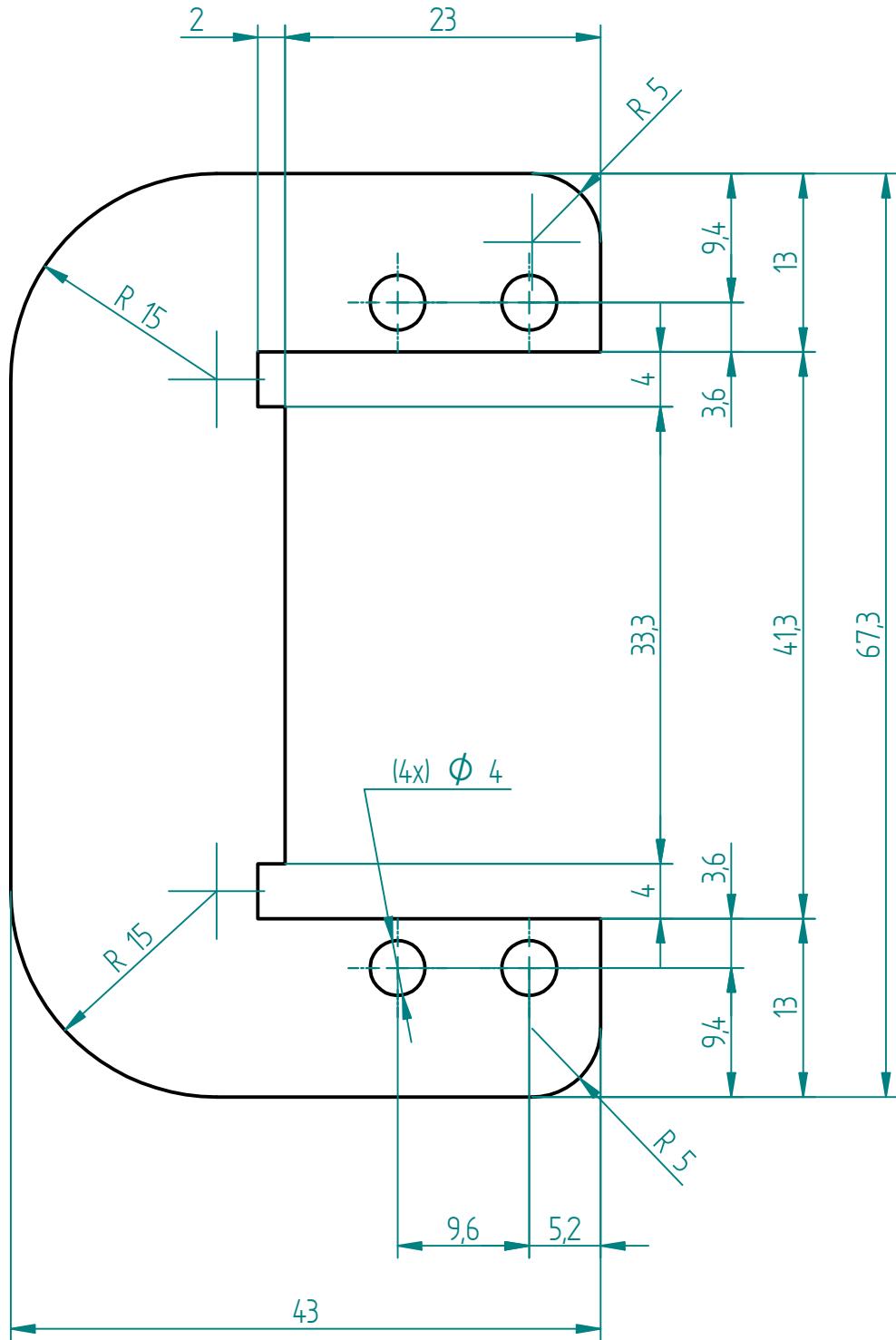


	NOMBRE	FECHA
DIBUJADO	M.I.G	12-02-15
MATERIAL	TRESPA	
ESPESOR	10 MM	
CANTIDAD	1 UNIDADES	
DIMENSIONES ACOTADAS EN MILIMETROS		
TOLERANCIA GENERAL $\pm 0.2$		
ANGULO $\pm 1^\circ$		

Proyecto Eurobot 2015

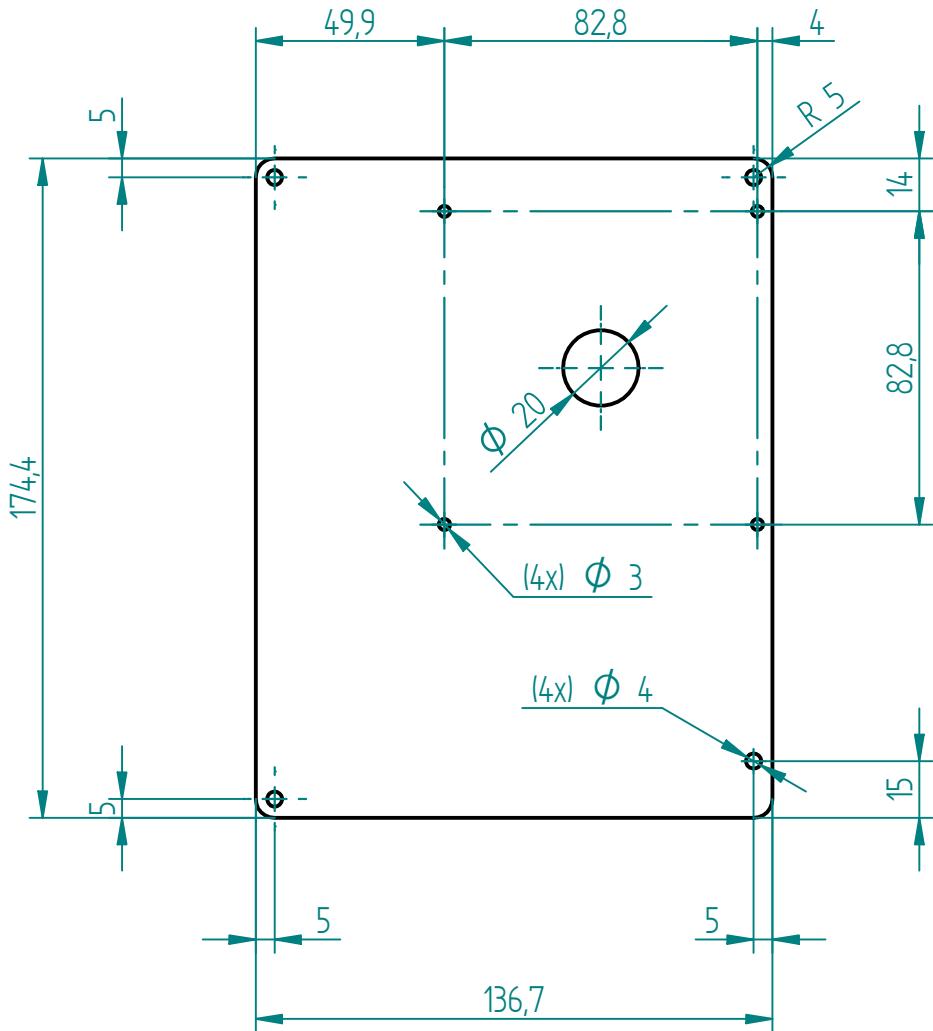
TITULO	BISAGRA TRAMPILLA SUPERIOR	
HOJA	NUMERO PLANO	REV
A4	12-00-03-077	0
ESCALA: 2:1	PESO:	HOJA 1 DE 1





NOMBRE	M.I.G	FECHA	Proyecto Eurobot 2015		
DIBUJADO					
MATERIAL	TRESPA				
ESPESOR	3 MM				
CANTIDAD	1 UNIDADES		TITULO SOporte SERVO TRAMPILLA SUPERIOR		
DIMENSIONES ACOTADAS EN MILIMETROS TOLERANCIA GENERAL ±0.2 ANGULO ±1°			HOJA	NUMERO PLANO	REV
			A4	12-00-04-201	0
			ESCALA:	PESO:	HOJA 1 DE 1



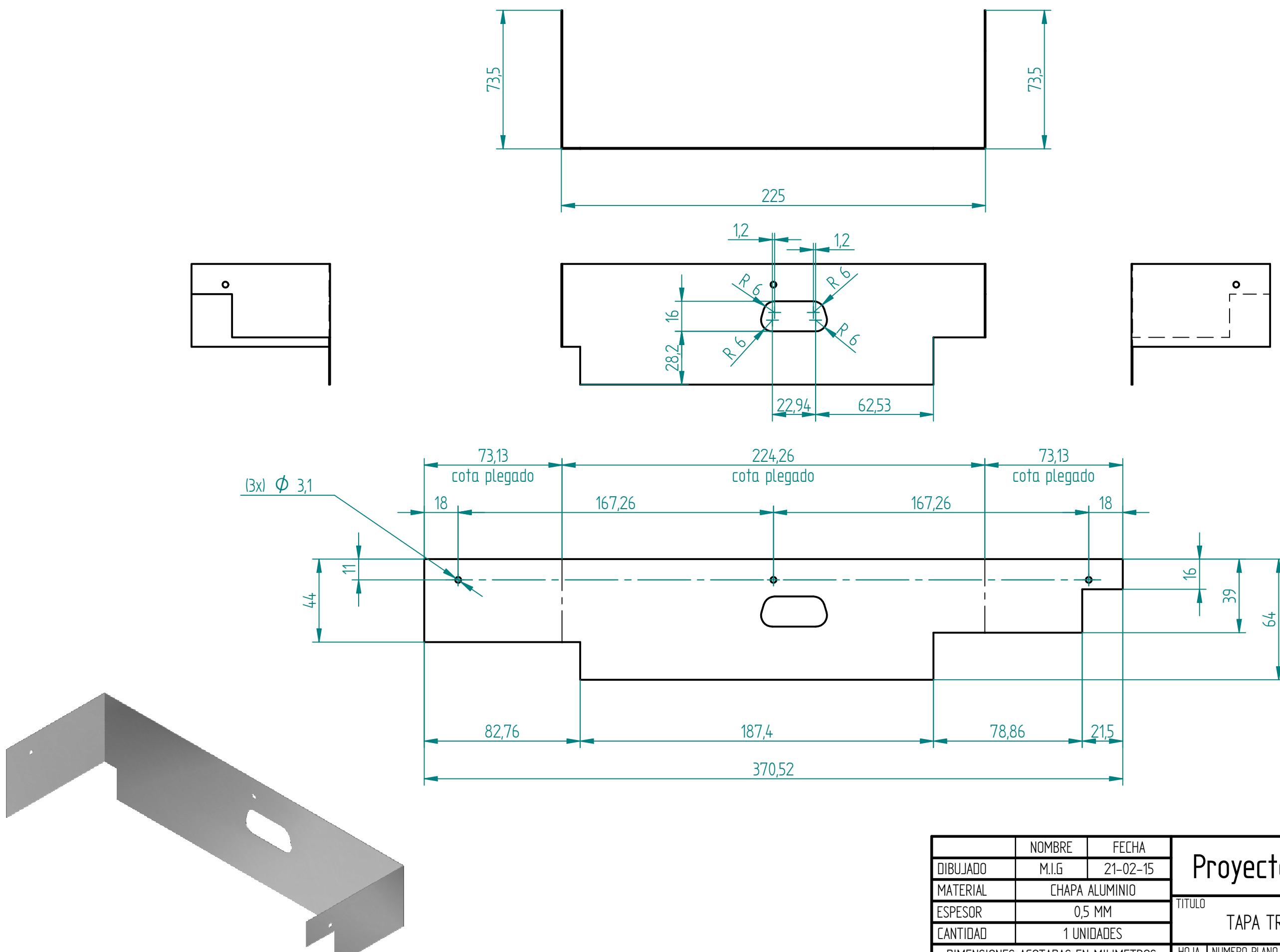


	NOMBRE	FECHA
DIBUJADO	M.I.G	18-02-15
MATERIAL	TRESPA	
ESPESOR	6 MM	
CANTIDAD	1 UNIDADES	
DIMENSIONES ACOTADAS EN MILIMETROS		
TOLERANCIA GENERAL $\pm 0.2$		
ANGULO $\pm 1^\circ$		

Proyecto Eurobot 2015

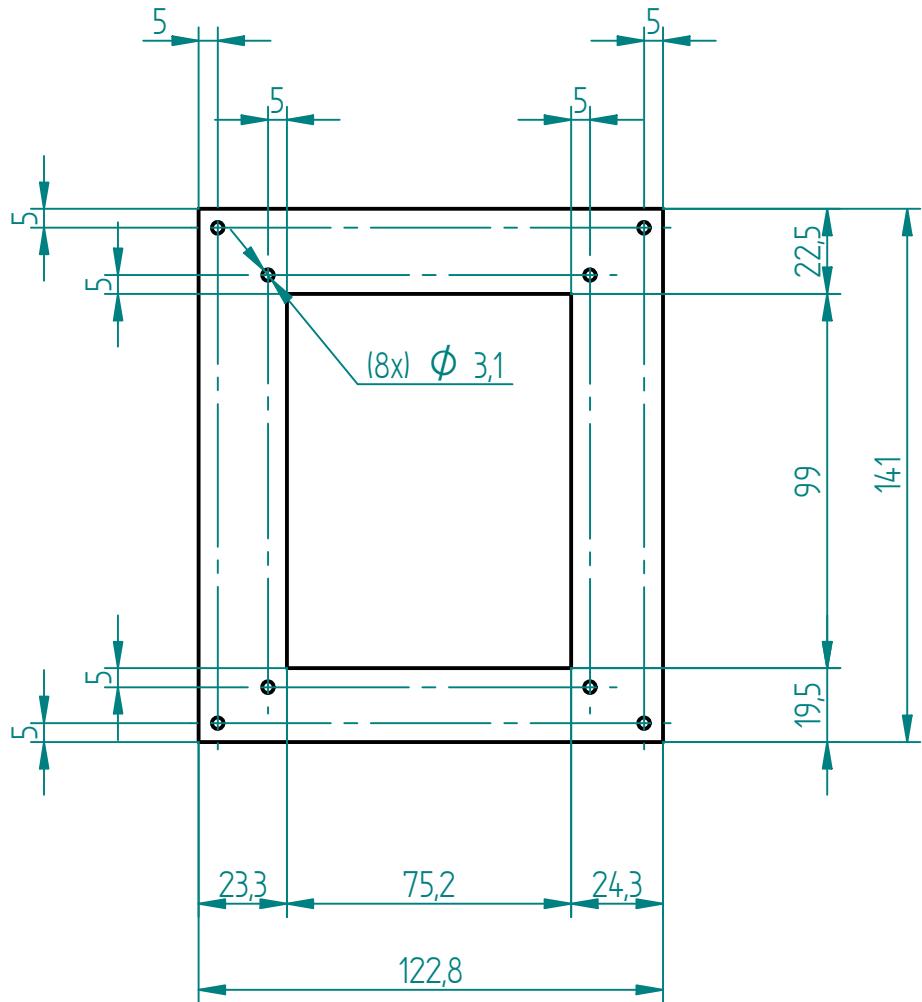
TITULO	TAPA ELECTRONICA	
HOJA	NUMERO PLANO	REV
A4	14-00-01-015	0
ESCALA: 1:2	PESO:	HOJA 1 DE 1





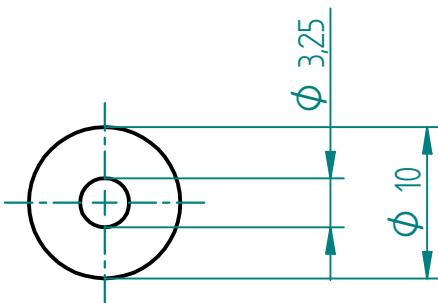
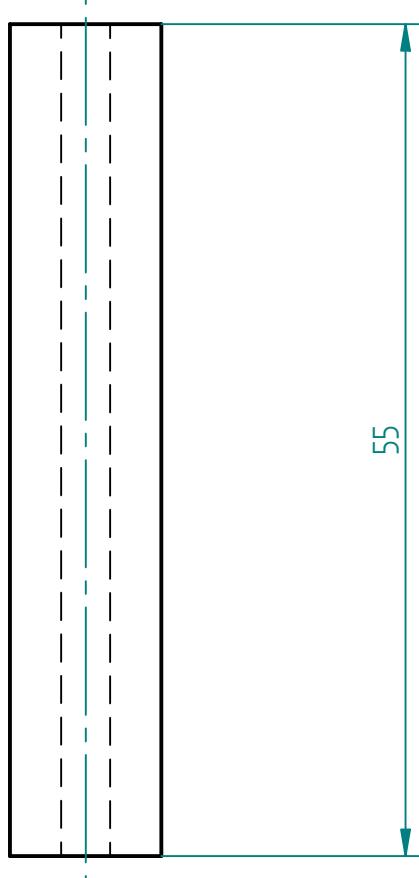
	NOMBRE	FECHA	Proyecto Eurobot 2015			
DIBUJADO	M.I.G	21-02-15				
MATERIAL	CHAPA ALUMINIO		TITULO			
ESPESOR	0,5 MM		TAPA TRASERA ELECTRONICA			
CANTIDAD	1 UNIDADES					
DIMENSIONES ACOTADAS EN MILIMETROS			HOJA	NUMERO PLANO		
TOLERANCIA GENERAL $\pm 0.2$			A3	14-00-02-018		
ANGULO $\pm 1^\circ$			REV	0		
ESCALA: 1:2		PESO:	HOJA 1 DE 1			





NOMBRE	M.I.G	FECHA	Proyecto Eurobot 2015		
DIBUJADO		18-02-15			
MATERIAL	TRESPA		TITULO		
ESPESOR	3 MM		SOPORTE ELECTRONICA		
CANTIDAD	1 UNIDADES				
DIMENSIONES ACOTADAS EN MILIMETROS			HOJA	NUMERO PLANO	REV
TOLERANCIA GENERAL ±0.2			A4	14-00-03-040	0
ANGULO ±1°			ESCALA:	PESO:	HOJA 1 DE 1





NOMBRE	FECHA	Proyecto Eurobot 2015		
DIBUJADO	M.I.G	18-02-15		
MATERIAL	ALUMINIO DURAL			TITULO
ESPESOR				SEPARADOR SOPORTE ELECTRONICA
CANTIDAD	4 UNIDADES			
DIMENSIONES ACOTADAS EN MILIMETROS		HOJA	NUMERO PLANO	REV
TOLERANCIA GENERAL $\pm 0.2$		A4	14-00-04-078	0
ANGULO $\pm 1^\circ$		ESCALA: 2:1	PESO:	HOJA 1 DE 1



# PLIEGO DE CONDICIONES

El objeto del presente pliego de condiciones es la descripción de las condiciones generales y particulares que han de regir cada una de las fases de este proyecto para garantizar la correcta consecución de su objetivo.

Se van a diferenciar dos fases principales, distinguiendo entre la etapa de diseño y la de ejecución material. Ambas fases podrían tener su propio pliego de condiciones de ejecución ya que en el desarrollo global del proyecto por un lado se da la relación entre el equipo de robótica que deriva una parte del proyecto bajo una especificaciones de diseño, pudiéndose considerar el equipo como el promotor del proyecto y yo el contratista que diseño bajo un pliego de condiciones, y por otro una vez concluida la fase de diseño y generada la documentación de fabricación, yo asumiría el rol de promotor del proyecto sobre las empresas que acometerían la fabricación y que serían las contratistas bajo el pliego de condiciones del proyecto de fabricación.

# 1. Pliego de condiciones de diseño

---

En este apartado se detallan las condiciones impuestas por el equipo de competición Eurobotics Engineering para el desarrollo de la mecánica y su funcionamiento.

## 1.1. Condiciones de diseño mecánico

---

Las condiciones de diseño han sido detalladas en el apartado 2.1Especificaciones de diseño. En este apartado se han especificado cada uno de los requerimientos mecánicos, incluyendo velocidades, aceleraciones, etc.

A estas condiciones de diseño hay que añadir las propias que unifican el diseño de los robots participantes en la prueba de Eurobot y que se encuentran recogidas en la normativa de la competición (Eurobot, 2014). Se ha incluido la normativa en el soporte digital del proyecto en la ruta:

[DOCUMENTOS TÉCNICOS\NORMATIVA COMPETICIÓN\E2015\\_Rules\\_EU\\_EN\\_final.pdf](#)

La normativa solo se encuentra disponible en lengua inglesa.

Dentro de este documento los apartados con las condiciones del diseño de los robots participantes son:

Capítulo E. “The robots” Los robots (pág. 19)

1. “Foreword” Prefacio (pág. 19)

Se explican las consideraciones generales como el número máximo de robots por equipo, que está limitado a dos.

2. “Dimensions” Dimensiones (pág. 20)

Se detallan las dimensiones máximas de la envolvente de cada uno de los dos robots del equipo, en el estado de arranque y en el estado posterior al arranque de máximo despliegue.

3. “Energy sources” Fuentes de energía (pág. 20)

Se detallan las fuentes de energías utilizables y las medidas de seguridad para cada una de ellas.

4. “Design constraints and mandatory equipment” Restricciones de diseño y equipamiento obligatorio (pág. 21)

En este apartado se enumera el equipamiento obligatorio y su disposición en los robots:

A. Cordón de arranque.

B. Botón de parada de emergencia.

C. Apagado automático obligatorio tras los 90 seg.

D. Sistema para evitar obstáculos.

E. En este apartado se detalla el tamaño y peso mínimo del obstáculo que los robots deben de ser capaces de evitar.

## 1.2. Condiciones de diseño software

---

El control de los actuadores, motores y sensores necesarios para la aplicación deberán ser controlados por el microcontrolador esclavo “IC5” dsPIC33FJ128MC804 de la tarjeta “mainboard Eurobotics”.

La programación se realizará utilizando como base las librerías Aversive (Microb Technology, Olivier MATZ, 2010) que son utilizadas en el resto de microcontroladores. Se utilizarán las librerías portadas a los microcontroladores dsPIC de Microchip por Javier Baliñas Santos.

El resto de condiciones de diseño han sido detalladas en el apartado de especificaciones software.

## 1.3. Condiciones de seguridad del diseño

---

Las condiciones de seguridad son todas derivadas de la normativa de Eurobot 2015 (Eurobot, 2014).

Estas consideraciones se encuentran en los capítulos:

Capítulo E. “The robots” Los robots (pág. 19)

### 1. “Safety” Seguridad (pág.23)

Se explican las consideraciones generales relativas a la seguridad.

Todos los sistemas instalados en el robot deben respetar las normativas y requisitos establecidos por las leyes nacionales y europeas. En concreto, los sistemas utilizados deben cumplir con las regulaciones relacionadas con la seguridad y no deben poner en peligro a los participantes o al público ni durante los partidos ni en la zona de participantes.

En general los robots no deben tener ningún elemento sobresaliente o afilado que pueda causar daño o pueda ser peligroso.

Está prohibido el uso de productos líquidos, corrosivos, materiales pirotécnicos o seres vivos.

Las fuentes de alimentación no deben de superar los 48 V.

En esta normativa también se encuentra regulado el uso de láser, la intensidad máxima de leds y la presión máxima de sistemas neumáticos. (Eurobot, 2014)

## 1.4. Condiciones de materiales

---

En general los materiales estructurales utilizados para el diseño asegurarán la resistencia necesaria para una prueba de estas características en la que se pueden producir choques con los robots oponentes y el tiempo disponible para una posible reparación es muy limitado. Los

materiales estructurales utilizados también deberán presentar una buena relación densidad-resistencia.

Como condición de diseño serán utilizados una serie de componentes que son enumerados en la Tabla 1-1 *Componentes que deberán ser utilizados en el diseño*. Debido a las limitaciones de financiación del proyecto se reutilizaran componentes de anteriores proyectos.

Componente	Función	Cantidad Disponible	Especificaciones para el diseño
Sensor Omron E3Z-D61	Sensores para detección puntual del oponente	Ilimitada	
Motor brushless Dunkermotoren BG40x50 con reductora incorporada PLG42S	Motores para el sistema de tracción bidireccional	2	
Encoder RI41 Hengstler 3600 pulsos	Encoder para el sistema de posicionamiento odométrico	2	Deben ser colocados en el mismo eje que las ruedas de tracción.
Servo Digital Dynamixel AX12-A	Servos digitales de conexión serie.	Ilimitada	El Número máximo de servos analógicos en el micro esclavo es de tres.
Motor Maxon 2322.980- 52.235-200 24V 6W, reductora 14:1, Encoder 100 pulsos 2 canales cuadratura	Sin uso determinado	1	
Conjunto de Baliza Giratoria	Sistema de control del oponente	1	Se debe intentar colocar su eje de giro sobre el eje del sistema bidireccional de tracción. La base superior debe estar colocada a 510 mm sobre el suelo

Tabla 1-1 *Componentes que deberán ser utilizados en el diseño*

## 2. Pliego de condiciones técnicas de fabricación

---

En este apartado se detallan las condiciones impuestas a las empresas subcontratadas para la realización material del diseño implementado y las condiciones de ensamblado y prueba del robot acabado.

### 2.1. Condiciones generales a empresas subcontratadas

---

Todas las piezas diseñadas “a medida” para el proyecto van a ser fabricadas por empresas externas con el soporte de los planos constructivos realizados para tal fin y en el caso de fabricación por maquinarias de control numérico siempre serán acompañados por archivos CAD en formato dxf para facilitar y asegurar la correcta realización de las piezas diseñadas.

Estas empresas están obligadas a cumplir las especificaciones detalladas en cada plano según se detalla:

1. Material, según marca o características si se detallan.  
En caso de no especificarse características deberá tener las adecuadas para el proceso de fabricación.
2. Dimensiones, todas las medidas acotadas sin definir tolerancias o ajustes de montaje tendrán una desviación máxima de:  
 $\pm 0.2$  mm en acotados lineales y  $\pm 1^\circ$  en acotados angulares.  
Todas las medidas acotadas sin unidad están expresadas en milímetros.
3. Todas las piezas recibidas serán comprobadas mediante calibre digital para las medidas de menor precisión e interiores y micrómetro para los ejes con ajustes de mayor precisión. Todas las piezas que no cumplan las especificaciones dimensionales serán devueltas para su reposición.
4. Los cantos y aristas de las piezas deberán ser acabadas mediante medios abrasivos (limas para metal o lija grano fino para los materiales fenólicos) para evitar que puedan producir cortes o lesiones a personas.
5. Los planos sin acotación directa de espesor de pieza mantendrán la tolerancia general de  $\pm 0.2$  mm para dicho espesor indicado en el tipo de material.

Las empresas subcontratadas están obligadas a permitir la entrada del ingeniero en sus talleres y poner a su disposición los medios humanos y materiales que precise para el correcto control de los trabajos realizados y en curso.

Será responsabilidad de las empresas contratadas los perjuicios que se deriven del no cumplimiento por su parte de las leyes y reglamentaciones existentes en el país sobre materia

laboral y social durante la realización de los trabajos, así como la relativa a la seguridad en el trabajo.

## **2.2. Condiciones para el montaje**

---

El montaje se realizará con piezas completamente acabadas por lo que se limitará al ensamblado mediante tornillería o mediante uniones con tolerancias de ajuste.

Para las uniones mediante tornillería solo se utilizarán las herramientas precisas. Se evitará el uso de llaves inglesas o puntas de atornilladores que no correspondan a la especificación de la cabeza de cada tornillo empleado.

Para los acoplos de piezas con tolerancias de ajuste, principalmente ejes sobre rodamientos, previamente se comprobará nuevamente la correcta medida de ajuste con el micrómetro. Nunca se intenta introducir un eje sin realizar esta comprobación previa.

Para el montaje se lubricarán las dos partes a unir y serán introducidas según el tipo de ajuste desde manualmente hasta con la ayuda de una martillo con protección de Nylon.

Se prestará especial atención al par de apriete utilizado para los diferentes tornillos. Aunque no se utilizarán llaves ni atornilladores dinamométricos se evitará sobrepasar los ligeros aprietes de métricas tan pequeñas como las utilizadas.

Siempre que sea necesario se utilizarán guantes y gafas de protección.

## **2.3. Pruebas de funcionamiento**

---

Una vez concluida la fase de montaje y programación se realizarán pruebas de estrés, sometiendo a los sistemas a un número de maniobras que duplique las esperadas por partida, en un tiempo no superior a 90 segundos.

Esta maniobra se repetirá cada media hora al menos 4 veces.

Con esta prueba se intenta simular un funcionamiento con el doble de exigencia del esperado durante la competición.

# PRESUPUESTO

## 1. Introducción

---

Debido a las características de este proyecto, este apartado no se va a ajustar a la definición propia de un presupuesto, entendiéndose este como el cálculo anticipado del coste. En este proyecto se ha concluido la ejecución tanto de desarrollo como material, por lo tanto en este capítulo se recogerá el coste final del proyecto, pero con la particularidad de que todas las piezas fabricadas bajo plazo han sido suministradas por empresas patrocinadoras o que colaboran desinteresadamente para el equipo de competición “Eurobotics Engineering”.

Para evitar que este capítulo se aleje de su objetivo final dentro de un proyecto de este tipo, se ha solicitado a dichas empresas que valoren el coste de fabricación de cada una de las piezas, según su tarifa de trabajo indicando el coste por unidad.

Para el cálculo de los costes se ha supuesto que es un proyecto de I+D (investigación y desarrollo), por lo que no se han valorado todos aquellos costes propios de un proyecto clásico o industrial, aunque si se ha mantenido la estructura clásica de un presupuesto con coste cero para los siguientes conceptos:

- Costes de certificación y visado del colegio de ingenieros.
- Honorarios profesionales de redacción de proyecto.

Dentro de las etapas de desarrollo del proyecto se han sucedido labores con diferentes exigencias de niveles de capacitación. Esto ha sido tenido en cuenta en el cálculo del coste de la mano de obra según la valoración de la Tabla 1-1 *Remuneración personal*.

Puesto	€/hora
Ingeniero Técnico Industrial Especialidad Electrónica Industrial	32
Técnico Superior Mecatrónica Industrial	15

Tabla 1-1 *Remuneración personal*

## 2. Presupuesto de ejecución material

---

Aunque este proyecto ha hecho uso del hardware de control para la conexión y la programación de los sistemas de manipulación de elementos de juego, la electrónica no va a ser incluida en el presupuesto al depender exclusivamente de otro de los integrantes del equipo.

Por lo tanto el presupuesto de ejecución material va a quedar restringido a todo el sistema mecánico desarrollado.

La ejecución material se va a dividir en los siguientes apartados:

- Presupuesto material.  
Se va a organizar por conjuntos y subconjuntos del sistema.
- Gastos generales.  
Se va a calcular el coste de amortización de los equipos y software utilizados exclusivamente para el desarrollo de ingeniería.
- Coste de personal.  
Este coste se va a dividir por las fases del proyecto, según la cualificación necesaria.

## 2.1. Coste de material

Se ha optado por organizar el listado de todos los elementos por conjuntos y subconjuntos del diseño, manteniendo de forma ordenada ascendente la codificación de los planos para poder localizar fácilmente todas las piezas.

En la Tabla 2-1 *Coste de materiales* se han incluido todos los componentes del diseño mecánico objeto de este proyecto y por lo tanto se han excluido todas las tarjetas electrónicas de control, baterías de alimentación, seta de emergencia y todos los componentes mecánicos y electrónicos del conjunto de la baliza de control de posición del oponente.

La tornillería que tampoco ha sido incluida, para no dificultar la visión general de la Tabla 2-1 *Coste de materiales*, ha sido detallada en la Tabla 2-2 *Coste de tornillería*.

Todos los precios codificados son sin IVA y tampoco se han incluido los gastos de transporte, que han sido especificados por proveedor en la Tabla 2-3 *Coste de portes*.

Conjunto	Sub Conjunto	Nº Plano	Descripción	Material	Cantidad	Pvp Unidad	Pvp Total	Proveedor
Robot	Estructura	00-01-01-001	Base 1	Trespa 6	1	30,45	30,45	Carpintería Ingles
Robot	Estructura	00-01-02-014	Tapa Superior Estructura	Trespa 6	1	27,3	27,3	Carpintería Ingles
Robot	Estructura	00-01-03-007	Estructura Vertical Delantero	Trespa 3	1	8,75	8,75	Carpintería Ingles
Robot	Estructura	00-01-04-008	Estructura Vertical Trasero	Trespa 3	1	9,36	9,36	Carpintería Ingles
Robot	Estructura	00-01-05-010	Estructura Lateral Izquierdo	Trespa 3	1	5,4	5,4	Carpintería Ingles
Robot	Estructura	00-01-06-011	Estructura Lateral Derecho	Trespa 3	1	6,3	6,3	Carpintería Ingles
Robot	Estructura	00-01-07-025	Estructura Tapa Bloque Motor	Trespa 10	1	31,86	31,86	Carpintería Ingles
Robot	Estructura	00-01-08-036	Estructura Soporte Brazos Palomitas	Trespa 3	1	2,7	2,7	Carpintería Ingles
Robot	Estructura	00-01-09-038	Estructura Soporte Brazos Palomitas Derecho	Trespa 3	1	2,88	2,88	Carpintería Ingles
Robot	Estructura	00-01-10-084	Tacos Aluminio	Macizo Aluminio 10x10	23	3,83	87,86	Ayudas Hidráulicas S.L.
Robot	Estructura	00-01-11-003	Refuerzo Delantero Base	Acero F-100	1	18,33	18,33	Ayudas Hidráulicas S.L.
Robot	Estructura	00-01-12-012	Estructura Chaflán Trasero	Trespa 3	1 S/D + 1 M/C	2,16	4,32	Carpintería Ingles
Robot	Estructura	00-01-13-027	Estructura Piso Trasero	Trespa 3	1	4,14	4,14	Carpintería Ingles

## Presupuesto

Robot	Estructura	00-01-14-026	Estructura Piso Trasero 1	Trespa 3	1	4,14	4,14	Carpintería Ingles
Robot	Estructura	00-01-15-028	Estructura Pared Vaso	Trespa 3	2	1,98	3,96	Carpintería Ingles
Robot	Estructura	00-01-16-029	Estructura Pendiente Piso Trasero 1	Trespa 3	2	1,98	3,96	Carpintería Ingles
Robot	Estructura	00-01-17-089	Palanca Apertura Trampillas Inferiores	Trespa 3	2	1,8	3,6	Carpintería Ingles
Robot	Estructura	00-01-18-203	Soporte Teflón Apoyo Trasero	Trespa 6	2	1,98	3,96	Carpintería Ingles
Robot	Estructura	00-01-19-054	Soporte Trasero Base Carros	Aluminio Dural Ø10	4	3,53	14,12	Ayudas Hidráulicas S.L.
Robot	Estructura	00-01-20-055	Soporte Lastre Y Base Carros Delantero	Aluminio Dural Ø10	4	2,82	11,28	Ayudas Hidráulicas S.L.
Robot	Estructura	00-01-21-053	Soporte Base Y Base Carros Delantero	Aluminio Dural Ø9	2	2,96	5,92	Ayudas Hidráulicas S.L.
Robot	Estructura	00-01-22-050	Soporte Rollon	Trespa 16	1 S/D + 1 M/C	57,6	115,2	Carpintería Ingles
Robot	Estructura		Unidad De Transferencia De Bola (Rollon) Altura: 19,6mm, Diámetro: 20,6mm Diámetro de Bola: 12,7mm, Peso de Carga Máx.:35kg (Ref. Farnell 1779545)		2	9,68	19,36	Farnell Components SL.
Robot	Cubierta	00-02-01-013	Trampilla Trasera Inferior	Chapa Aluminio 0,5mm	1 S/D + 1 M/C	34,24	68,48	Ayudas Hidráulicas S.L.
Robot	Cubierta	00-02-02-017	Trampilla Trasera Superior	Chapa Aluminio 0,5mm	1	26,68	26,68	Ayudas Hidráulicas S.L.
Robot	Cubierta		Bisagra Recta Hierro Cincado 25x18 mm		6	0,54	3,24	Leroy Merlin
Robot	Cubierta		Cinta de Raso ancho 10 mm Negra		0,5	1,6	0,8	Estilo pontejos
Robot	Electrónica		Sensor Omron E3Z-D61		9	15,84	142,56	EBAY
TOTAL ESTRUCTURA ROBOT							666,91 €	
Bloque Motor	Estructura	01-00-01-000	Placa Encoder Izquierdo	Aluminio 5083 8mm	2	78,64	157,28	Ayudas Hidráulicas S.L.
Bloque Motor	Estructura	01-00-02-000	Placa Motor Izquierda	Aluminio 5083 8mm	2	78,64	157,28	Ayudas Hidráulicas S.L.
Bloque Motor	Estructura	01-00-03-000	Llanta	Aluminio Dural	2	45,4	90,8	Ayudas Hidráulicas S.L.
Bloque Motor		01-00-04-000	Rueda Foam Ancho 40	Tubo Foam 7,5mm	2	4,55	9,1	Leroy Merlin
Bloque Motor	Estructura		Rodamiento GPZ Ext. Ø26, Int. Ø10, Esp. 8mm		4	1,68	6,72	World Bearing S.L.
Bloque Motor	Electrónica		Motor brushless Dunkermotoren BG40x50 con reductora incorporada PLG42S		2	100	200	EBAY
Bloque Motor	Transmisión		POLEA DENTADA PASO 5MM 30 DIENTES	Aluminio	2	6,4	12,8	World Bearing S.L.
Bloque Motor	Transmisión		POLEA DENTADA PASO 5MM 15 DIENTES	Aluminio	2	4,2	8,4	World Bearing S.L.
Bloque Motor	Transmisión		CORREA DENTADA SYNCHROBELT PASO 5MM 225 MM	Poliuretano armada con Acero	2	5,2	10,4	World Bearing S.L.
TOTAL BLOQUE MOTOR							652,78 €	
Lastre	Estructura	02-00-01-006	Trasero	Acero	1	30,3	30,3	Ayudas Hidráulicas S.L.
Lastre	Estructura	02-00-02-006	Trasero Superior	Acero	1	45,2	45,2	Ayudas Hidráulicas S.L.
Lastre	Estructura	02-00-03-005	Delantero	Acero F-100	2	19,8	39,6	Ayudas Hidráulicas S.L.
Lastre	Estructura	02-00-04-204	Suplemento Trasero	Trespa 3	1	2,34	2,34	Carpintería Ingles
TOTAL LASTRE							117,44 €	
Balancín Encoder	Estructura	03-00-01-030	Bisagra Exterior Balancín Encoder	Trespa 3	2	2,52	5,04	Carpintería Ingles
Balancín Encoder	Estructura	03-00-02-031	Bisagra Interior Balancín Encoder	Trespa 3	2	9,36	18,72	Carpintería Ingles

Balancín Encoder	Estructura	03-00-03-032	Brazo Exterior Balancín Encoder	Trespa 3	2	2,7	5,4	Carpintería Ingles
Balancín Encoder	Estructura	03-00-04-033	Brazo Interior Balancín Encoder	Trespa 3	2	2,52	5,04	Carpintería Ingles
Balancín Encoder	Estructura	03-00-05-000	Rueda Balancín Encoder	Aluminio Dural Ø53	2	35,53	71,06	Ayudas Hidráulicas S.L.
Balancín Encoder	Estructura	03-00-06-065	Separador Brazos Encoder	Aluminio Dural Ø10	8	3,53	28,24	Ayudas Hidráulicas S.L.
Balancín Encoder	Estructura	03-00-07-066	Eje Brazos Encoder	Aluminio Dural Ø14	8	9,82	78,56	Ayudas Hidráulicas S.L.
Balancín Encoder	Estructura		Rodamiento miniatura Ext. Ø15, Int. Ø6, Esp. 5mm (12696-ZZ)		4	1,99	7,96	World Bearing S.L.
Balancín Encoder	Estructura		Encoder RI41 Hengstler 3600 pulsos		2	200	400	EBAY
TOTAL BALANCIN ENCODER							620,02 €	
Conjunto Carro 2	Estructura	04-00-01-009	Base Carros Izquierda	Trespa 6	1	16,2	16,2	Carpintería Ingles
Conjunto Carro 2	Estructura	04-00-02-093	Tapa Bisagra Superior	Trespa 6	2	21,96	43,92	Carpintería Ingles
Conjunto Carro 2	Estructura	04-00-03-094	Guías 6 Stand	Varilla Acero Plata Ø6	4	14,5	58	Ayudas Hidráulicas S.L.
Conjunto Carro 2	Estructura	04-00-04-095	Guías 8 Carros	Varilla Acero Plata Ø8	4	18,9	75,6	Ayudas Hidráulicas S.L.
Conjunto Carro 2	Carro Pinza	04-01-01-106	Base 1 Carro Pinza	Trespa 3	2	2,7	5,4	Carpintería Ingles
Conjunto Carro 2	Carro Pinza	04-01-02-107	Base 3 Carro Pinza	Trespa 3	2	3,24	6,48	Carpintería Ingles
Conjunto Carro 2	Carro Pinza	04-01-03-108	Base 2 Carro Pinza	Trespa 3	2	2,52	5,04	Carpintería Ingles
Conjunto Carro 2	Carro Pinza	04-01-04-109	Base 4 Carro Pinza	Trespa 10	1 S/D + 1 M/C	35,1	70,2	Carpintería Ingles
Conjunto Carro 2	Carro Pinza	04-01-05-110	Pinza Suplemento	Trespa 6	8	22,32	178,56	Carpintería Ingles
Conjunto Carro 2	Carro Pinza	04-01-06-111	Pinza Palanca	Trespa 3	4	2,34	9,36	Carpintería Ingles
Conjunto Carro 2	Carro Pinza	04-01-07-112	Pinza Leva	Trespa 3	8	1,8	14,4	Carpintería Ingles
Conjunto Carro 2	Carro Pinza	04-01-08-113	Cabeza Servo Carro Pinza	Trespa 3	2	2,52	5,04	Carpintería Ingles
Conjunto Carro 2	Carro Pinza	04-01-09-114	Alojamiento Eje Servo	Trespa 6	2	4,36	8,72	Carpintería Ingles
Conjunto Carro 2	Carro Pinza	04-01-10-115	Separador Lateral Base 1 Carro Pinza	Aluminio Dural Ø9	4	2,96	11,84	Ayudas Hidráulicas S.L.
Conjunto Carro 2	Carro Pinza	04-01-11-116	Separador Central Base 1 Carro Pinza	Aluminio Dural Ø12	4	4,15	16,6	Ayudas Hidráulicas S.L.
Conjunto Carro 2	Carro Pinza	04-01-12-117	Separador Lateral Base 2 Carro Pinza	Aluminio Dural Ø9	4	1,48	5,92	Ayudas Hidráulicas S.L.
Conjunto Carro 2	Carro Pinza	04-01-13-118	Separador Central Base 2 Carro Pinza	Aluminio Dural Ø12	2	2,08	4,16	Ayudas Hidráulicas S.L.
Conjunto Carro 2	Carro Pinza	04-01-14-119	Separador Lateral Base 3 Carro Pinza	Aluminio Dural Ø9	4	5,92	23,68	Ayudas Hidráulicas S.L.
Conjunto Carro 2	Carro Pinza	04-01-15-120	Separador Central Base 3 Carro Pinza	Aluminio Dural Ø12	2	8,3	16,6	Ayudas Hidráulicas S.L.
Conjunto Carro 2	Carro Pinza		Futaba S3003 Servo 180°		2	8,22	16,44	PC COMPONENTES Y MULTIMEDIA SL
Conjunto Carro 2	Carro Pinza		Rodamiento lineal Ext. Ø15 , Int. Ø8, Longitud 24mm LM8UU 8mm Linear Bearing	Acero al carbono	8	0,93	7,44	(AliExpress) Tigoo Technology Co., Ltd. China (Mainland)
Conjunto Carro 2	Carro Pinza		Casquillos sinterizados en bronce autolubricado TIPO B Ext. Ø8, Int. Ø4, Valona Ø12, Longitud 4 mm, Espesor Valona 2 mm	Bronce sinterizado	8	0,85	6,8	Maymol S.A.

## Presupuesto

Conjunto Carro 2	Carro Pinza		Casquillos sinterizados en bronce autolubricado TIPO B Ext. Ø6, Int. Ø3, Valona Ø9, Longitud 4 mm, Espesor Valona 1,5 mm	Bronce sinterizado	2	1	2	Maymol S.A.
Conjunto Carro 2	Carro Servo	04-02-01-096	Palanca Carro Servo	Trespa 3	4	1,98	7,92	Carpintería Ingles
Conjunto Carro 2	Carro Servo		Casquillos sinterizados en bronce autolubricado TIPO B Ext. Ø6, Int. Ø3, Valona Ø9, Longitud 4 mm, Espesor Valona 1,5 mm	Bronce sinterizado	4	1	4	Maymol S.A.
Conjunto Carro 2	Carro Servo	04-02-02-097	Plato Carro Servo Lado Soporte Eje	Trespa 3	2	1,98	3,96	Carpintería Ingles
Conjunto Carro 2	Carro Servo	04-02-03-098	Plato Carro Servo Lado Servo	Trespa 3	2	2,16	4,32	Carpintería Ingles
Conjunto Carro 2	Carro Servo	04-02-04-099	Separador Platos Carro Servo	Trespa 5	2	1,1	2,2	Carpintería Ingles
Conjunto Carro 2	Carro Servo	04-02-05-100	Tapa Carro Servo Derecha_Mir1	Trespa 10	1 S/D + 1 M/C	35,1	70,2	Carpintería Ingles
Conjunto Carro 2	Carro Servo	04-02-06-101	Base Carro Servo Derecha_Mir1	Trespa 10	1 S/D + 1 M/C	35,1	70,2	Carpintería Ingles
Conjunto Carro 2	Carro Servo	04-02-07-102	Soporte Servo Carro Servo	Trespa 3	2	4,5	9	Carpintería Ingles
Conjunto Carro 2	Carro Servo	04-02-08-103	Soporte Eje Servo Carro Servo	Trespa 3	2	3,6	7,2	Carpintería Ingles
Conjunto Carro 2	Carro Servo	04-02-09-104	Separador Lateral Carro Servo	Aluminio Dural Ø12	4	15,66	62,64	Ayudas Hidráulicas S.L.
Conjunto Carro 2	Carro Servo	04-02-10-105	Separador Central Carro Servo	Aluminio Dural Ø14	2	21,64	43,28	Ayudas Hidráulicas S.L.
Conjunto Carro 2	Carro Servo		Rodamiento miniatura Ext. Ø10, Int. Ø4, Esp. 4mm (12MR-104-ZZ)		2	1,88	3,76	World Bearing S.L.
Conjunto Carro 2	Carro Servo		Rodamiento lineal Ext. Ø15, Int. Ø8, Longitud 24mm LM8UU 8mm Linear Bearing	Acero al carbono	8	0,93	7,44	(AliExpress) Tigoo Technology Co., Ltd. China (Mainland)
Conjunto Carro 2	Carro Servo		Servo Digital Dynamixel AX12-A		2	46,28	92,56	RO-BOTICA GLOBAL S.L.
"TOTAL CONJUNTO CARRO 2" 997,08 €								
Paletas Stand	Estructura	05-00-01-002	Inferior Estructura Paletas	Trespa 6	1	5,04	5,04	Carpintería Ingles
Paletas Stand	Estructura	05-00-02-034	Medio Estructura Paletas	Trespa 3	1	2,16	2,16	Carpintería Ingles
Paletas Stand	Estructura	05-00-03-035	Superior Estructura Paletas	Trespa 3	1	4,86	4,86	Carpintería Ingles
Paletas Stand	Estructura	05-00-04-056	Separador Estructura Ti	Aluminio Dural Ø14	1	9,57	9,57	Ayudas Hidráulicas S.L.
Paletas Stand	Estructura	05-00-05-057	Separador Estructura Ts	Aluminio Dural Ø14	1	4,95	4,95	Ayudas Hidráulicas S.L.
Paletas Stand	Estructura	05-00-06-058	Separador Estructura Di	Aluminio Dural Ø18	1	15,08	15,08	Ayudas Hidráulicas S.L.
Paletas Stand	Estructura	05-00-07-059	Separador Estructura Ds	Aluminio Dural Ø18	1	7,8	7,8	Ayudas Hidráulicas S.L.
Paletas Stand	Estructura	05-00-08-062	Brida Paletas	Aluminio Dural Ø26	1	26,6	26,6	Ayudas Hidráulicas S.L.
Paletas Stand	Estructura	05-00-09-063	Paleta	Trespa 3	2	2,7	5,4	Carpintería Ingles
Paletas Stand	Estructura		Servo Digital Dynamixel AX12-A		2	46,28	92,56	RO-BOTICA GLOBAL S.L.
Paletas Stand	Transmisión		POLEA DENTADA PASO 2,5MM 30 DIENTES (1016T2.5-Z-30)	Aluminio	2	4,65	9,3	World Bearing S.L.
Paletas Stand	Transmisión		CORREA DENTADA PASO 2,5MM 177,5MM (26T-2.5-177.5-6)	Poliuretano armada con Acero	1	3,64	3,64	World Bearing S.L.
TOTAL PALETAS STAND 186,96 €								
Carro Horizontal	Estructura		Guía Igus "DryLin T" Ts-04-09 Longitud 324 mm	Aluminio Tratado	0,32	27,07	8,77	Igus Polymer Innovations, S.L. Unipersonal
Carro Horizontal	Estructura		Corte a medida de "Guía Igus "DryLin T" Ts-04-09"		1	1	1	Igus Polymer Innovations, S.L. Unipersonal

Carro Horizontal	Estructura	06-00-01-004	Guía Carro Horizontal (mecanizado guía Igus)	Guía Igus Ts-04-09	1	6,2	6,2	Ayudas Hidráulicas S.L.
Carro Horizontal	Estructura	06-00-02-051	Brazo Carro Horizontal	Trespa 3	1	2,7	2,7	Carpintería Ingles
Carro Horizontal	Estructura	06-00-03-067	Soporte Motor Carro Horizontal	Trespa 6	1	23,04	23,04	Carpintería Ingles
Carro Horizontal	Estructura	06-00-04-070	Tensor Inferior Correa	Trespa 10	2	32,4	64,8	Carpintería Ingles
Carro Horizontal	Estructura		Carro deslizante TW-04-09		1	9,5	9,5	Igus Polymer Innovations, S.L. Unipersonal
Carro Horizontal	Transmisión		Rodamiento miniatura Ext Ø10, Int. Ø4, Esp. 4mm (12MR-104-ZZ)		2	1,88	3,76	World Bearing S.L.
Carro Horizontal	Transmisión		POLEA DENTADA PASO 2,5MM 12 DIENTES (1016T2,5-Z-12)	Aluminio	2	4,47	8,94	World Bearing S.L.
Carro Horizontal	Transmisión		CORREA DENTADA PASO 2,5MM 650MM (26T-2,5-650-6)	Poliuretano armada con Acero	1	5,56	5,56	World Bearing S.L.
Carro Horizontal	Transmisión		Motor Maxon 2322,980-52,235-200 24V 6W, reductora 14:1, Encoder 100 pulsos 2 canales cuadratura		1	77	77	EBAY
TOTAL CARRO HORIZONTAL							211,27 €	
Puerta Stand	Estructura	07-00-01-020	Base Puerta Inferior	Trespa 6	1	6,12	6,12	Carpintería Ingles
Puerta Stand	Estructura	07-00-02-047	Brazo Puerta Superior	Trespa 3	4	2,7	10,8	Carpintería Ingles
Puerta Stand	Estructura	07-00-03-048	Puerta	Trespa 3	2	2,34	4,68	Carpintería Ingles
Puerta Stand	Estructura	07-00-04-079	Leva Servo Puerta	Trespa 3	2	1,98	3,96	Carpintería Ingles
Puerta Stand	Estructura	07-00-05-200	Soporte Servo Inferior	Trespa 3	1	3,06	3,06	Carpintería Ingles
Puerta Stand	Estructura		Rodamiento GPZ Ext. Ø19, Int. Ø8, Esp. 6mm (01626-ZZ)		4	0,85	3,4	World Bearing S.L.
Puerta Stand	Estructura		Cordon elástico Ø3 Negro		1 m	0,8	0,8	Estilo pontejos
Puerta Stand	Estructura		Servo Digital Dynamixel AX12-A		2	46,28	92,56	RO-BOTICA GLOBAL S.L.
TOTAL PUERTA STAND							125,38 €	
Pinza Vaso Delantero	Estructura	08-00-01-021	Rampa Base	Trespa 3	1	3,42	3,42	Carpintería Ingles
Pinza Vaso Delantero	Estructura	08-00-02-022	Soporte Dynamixel Fijo Lateral	Trespa 3	1	4,68	4,68	Carpintería Ingles
Pinza Vaso Delantero	Estructura	08-00-03-023	Soporte Dynamixel Fijo Lateral Derecho	Trespa 3	1	3,78	3,78	Carpintería Ingles
Pinza Vaso Delantero	Estructura	08-00-04-024	Soporte Dynamixel Fijo	Trespa 3	1	3,96	3,96	Carpintería Ingles
Pinza Vaso Delantero	Estructura		Servo Digital Dynamixel AX12-A		1	46,28	46,28	RO-BOTICA GLOBAL S.L.
Pinza Vaso Delantero	Brazo	08-01-01-041	Apoyo Inferior	Trespa 3	1	1,8	1,8	Carpintería Ingles
Pinza Vaso Delantero	Brazo	08-01-02-042	Apoyo Superior	Trespa 3	1	2,16	2,16	Carpintería Ingles
Pinza Vaso Delantero	Brazo	08-01-03-043	Brazo Soporte Dynamixel Móvil	Trespa 3	1	3,24	3,24	Carpintería Ingles
Pinza Vaso Delantero	Brazo	08-01-04-044	Brazo Estabilizador Soporte Dynamixel Móvil	Trespa 3	1	1,8	1,8	Carpintería Ingles
Pinza Vaso Delantero	Brazo	08-01-05-045	Separador Dynamixel Móvil	Trespa 8	1	14,58	14,58	Carpintería Ingles
Pinza Vaso Delantero	Brazo	08-01-06-046	Pinza	Trespa 3	1	2,52	2,52	Carpintería Ingles
Pinza Vaso Delantero	Brazo		Servo Digital Dynamixel AX12-A		1	46,28	46,28	RO-BOTICA GLOBAL S.L.
TOTAL PINZA VASO DELANTERO							134,5 €	
Brazo Claqueta	Estructura	09-00-01-071	Disco Brazo Claqueta	Trespa 3	2 S/D + 2 M/C	1,98	3,96	Carpintería Ingles
Brazo Claqueta	Estructura	09-00-02-000	Varilla Brazo Claqueta	Fibra De Vidrio Ø2	2	3,2	6,4	Ayudas Hidráulicas S.L.

## Presupuesto

Brazo Claqueeta	Estructura		Futaba S3003 Servo 180º		2	8,22	16,44	PC COMPONENTES Y MULTIMEDIA SL
TOTAL BRAZO CLAQUETA 26,8 €								
Brazos Palomitas	Estructura	10-01-01-037	Estructura Soporte Servo Dynamixel Brazos Palomitas	Trespa 3	1	4,5	4,5	Carpintería Ingles
Brazos Palomitas	Estructura	10-01-02-039	Estructura Soporte Servo Dynamixel Brazos Palomitas Derecho	Trespa 3	1	19,26	19,26	Carpintería Ingles
Brazos Palomitas	Estructura	10-01-03-083	Base Brazos Palomitas 2	Trespa 3	2	3,42	6,84	Carpintería Ingles
Brazos Palomitas	Estructura	10-01-04-082	Pendiente Brazos Palomitas 2	Trespa 3	4	2,7	10,8	Carpintería Ingles
Brazos Palomitas	Estructura	10-01-05-081	Tapa Brazos Palomitas 2	Trespa 3	2	2,16	4,32	Carpintería Ingles
Brazos Palomitas	Estructura	10-01-06-085	Suplemento Servo Brazos Palomitas	Aluminio Dural Ø22	2	33,45	66,9	Ayudas Hidráulicas S.L.
Brazos Palomitas	Estructura		Servo Digital Dynamixel AX12-A		2	46,28	92,56	RO-BOTICA GLOBAL S.L.
TOTAL BRAZOS PALOMITAS 205,18 €								
Pinza Vaso	Estructura	11-00-01-086	Pinza Completa	Trespa 3	1	3,24	3,24	Carpintería Ingles
Pinza Vaso	Estructura	11-00-02-087	Pinza Partida Eje	Trespa 3	1	2,88	2,88	Carpintería Ingles
Pinza Vaso	Estructura	11-00-03-088	Pinza Partida Punta	Trespa 3	1	1,8	1,8	Carpintería Ingles
Pinza Vaso	Estructura		Servo Digital Dynamixel AX12-A		2	46,28	92,56	RO-BOTICA GLOBAL S.L.
TOTAL PINZA VASO 100,48 €								
Trampilla Superior	Estructura	12-00-01-075	Leva Servo Trampilla Superior 2	Trespa 3	1	1,8	1,8	Carpintería Ingles
Trampilla Superior	Estructura	12-00-02-076	Brazo Servo Trampilla Superior 2	Trespa 3	1	1,98	1,98	Carpintería Ingles
Trampilla Superior	Estructura	12-00-03-077	Bisagra Trampilla Superior	Trespa 10	1	31,32	31,32	Carpintería Ingles
Trampilla Superior	Estructura	12-00-04-201	Soporte Servo Trampilla Superior	Trespa 3	1	2,7	2,7	Carpintería Ingles
Trampilla Superior	Estructura		Futaba S3003 Servo 180º		1	8,22	8,22	PC COMPONENTES Y MULTIMEDIA SL
TOTAL TRAMPILLA SUPERIOR 46,02 €								
Electrónica	Estructura	14-00-01-015	Tapa Electrónica	Trespa 6	1	5,4	5,4	Carpintería Ingles
Electrónica	Estructura	14-00-02-018	Tapa Trasera Electrónica	Chapa Aluminio 0,5mm	1	24,42	24,42	Ayudas Hidráulicas S.L.
Electrónica	Estructura	14-00-03-040	Soporte Electrónica	Trespa 3	1	2,34	2,34	Carpintería Ingles
Electrónica	Estructura	14-00-04-078	Separador Soporte Electrónica	Aluminio Dural Ø10	4	8,25	33	Ayudas Hidráulicas S.L.
Electrónica			Bolsa de 10 cables de 4 pines de 20 cm para servos Dynamixel. (Ref. 2495)		1	22,23	22,23	RO-BOTICA GLOBAL S.L.
Electrónica			Bolsa de 10 cables de 4 pines de 24 cm para servos Dynamixel. (Ref. 2490)		1	23,97	23,97	RO-BOTICA GLOBAL S.L.
Electrónica			Bobina estafío 0,5 mm 250grs sin plomo		1	18,35	18,35	Farnell Components SL.
TOTAL SOPORTE ELECTRÓNICA 129,71 €								
<b>TOTAL 4220,53 €</b>								

Tabla 2-1 Coste de materiales

Din	Métrica	Longitud	Cantidad	Precio Ud.	Total	Código RS
84	M2	6	8	0,0282 €	0,23 €	526-906
84	M2	16	8	0,0597 €	0,48 €	482-9120
84	M2,5	6	2	0,0302 €	0,06 €	546-6102
84	M2,5	10	4	0,0316 €	0,13 €	546-6118
912	M3	12	4	0,1164 €	0,47 €	491-5209
912	M3	15	10	0,1919 €	1,92 €	822-9063
912	M3	16	4	0,1421 €	0,57 €	281-013
912	M4	12	8	0,1357 €	1,09 €	281-035
912	M4	14(20)	4	0,1120 €	0,45 €	227-6849
912	M4	18(20)	5	0,1120 €	0,56 €	227-6849
912	M4	25	2	0,1640 €	0,33 €	491-5237
912	M4	30	2	0,1654 €	0,33 €	483-8247
912	M4	35	4	0,2583 €	1,03 €	822-9072
912	M4	60	2	0,3567 €	0,71 €	ROME S.A.
963	M2	6	16	0,0341 €	0,55 €	482-7685
963	M2	8(12)	2	0,0400 €	0,08 €	482-9114
963	M2	10(12)	36	0,0400 €	1,44 €	482-9114
963	M2	12	28	0,0400 €	1,12 €	482-9114
963	M2	14(16)	8	0,0597 €	0,48 €	482-9120
7984	M6	16	10	0,1395 €	1,40 €	281-300
7985	M3	10(12)	8	0,0446 €	0,36 €	279-155
7985	M3	12	1	0,0446 €	0,04 €	279-155
7991	M3	4(6)	10	0,0708 €	0,71 €	304-4788
7991	M3	6	28	0,0708 €	1,98 €	304-4788
7991	M3	10	30	0,0651 €	1,95 €	232-8366
7991	M3	12	4	0,0734 €	0,29 €	232-8372
7991	M3	14(16)	8	0,1402 €	1,12 €	292-423
7991	M3	16	20	0,1402 €	2,80 €	292-423
7991	M4	12	24	0,0968 €	2,32 €	171-821
7991	M4	14(16)	24	0,1014 €	2,43 €	171-837
7991	M4	16	8	0,1014 €	0,81 €	171-837
7991	M4	20	12	0,1268 €	1,52 €	292-445
7991	M4	30	4	0,1598 €	0,64 €	491-5108
7991	M4	35	2	0,1753 €	0,35 €	ROME S.A.
7991	M4	45	2	0,2366 €	0,47 €	ROME S.A.
7991	M4	55(60)	4	0,3567 €	1,43 €	ROME S.A.
7991	M5	20	12	0,1148 €	1,38 €	171-865
GUSARO	M5	5	2	0,1454 €	0,29 €	ROME S.A.
Varilla	M4	95	1	0,5663 €	0,57 €	ROME S.A.
934	M2		100	0,0359 €	3,59 €	528-126
934	M2,5		6	0,0367 €	0,22 €	528-132
934	M3		125	0,0371 €	4,64 €	525-701
934	M4		100	0,0426 €	4,26 €	525-717
934	M5		14	0,0465 €	0,65 €	525-723
TOTAL					48,24 €	

Tabla 2-2 Coste de tornillería

Proveedor	Coste porte
Carpintería Ingles	0 €
Ayudas Hidráulicas S.L.	0 €
Farnell Components SL.	9 €
Leroy Merlin	0 €
Estilo pontejos	0 €
EBAY	35 €
World Bearing S.L.	0 €
PC COMPONENTES Y MULTIMEDIA S.L.	12 €
(AliExpress) Tigoo Technology Co., Ltd. China (Mainland)	15 €
Maymol S.A.	0 €
RO-BOTICA GLOBAL S.L.	4,96 €
Igus Polymer Innovations, S.L. Unipersonal	10 €
RS Amidata S.A.	5 €
Rome S.A.	0 €
<b>TOTAL</b>	<b>90,96 €</b>

Tabla 2-3 Coste de portes

## 2.2. Coste de mano de obra

Fase Proyecto	Cualificación	Horas	Coste/hora	Coste
Diseño mecánico	Ing. Técnico Industrial	160	32 €/h	5120 €
Generación de planos constructivos	Técnico Superior	32	15 €/h	480 €
Montaje mecánico	Técnico Superior	24	15 €/h	360 €
Adaptación de electrónica dedicada a sistemas de manipulación de elementos de juego	Ing. Técnico Industrial	8	32 €/h	256 €
Diseño software	Ing. Técnico Industrial	200	32 €/h	6400 €
<b>TOTAL</b>				<b>12.616 €</b>

Tabla 2-4 Coste de mano de obra por fases de proyecto

## 2.3. Coste de amortización de equipos y software

---

Equipo	Precio	Periodo amortización	Tiempo Proyecto	Amortización
PC Procesador Core i7-4790K CPU 4.00GHz 16.0GB RAM (Incluye componentes puesto completo)	1315 €	3 años	10 semanas	84 €
Programador Microchip MPLAB ICD 3 (Ref. DV164035)	188,46 €	5 años	5,7 semanas	2,06 €
Polímetro	120 €	5 años	1 semana	0,46 €
Soldador WELLER (Ref. 0056813199)	25,37 €	5 años	1 semana	0,09 €
Surtido de 55 herramientas con maleta polipropileno antichoques BETA Ref. 2051N	640 €	10 años	1 semana	1,23 €
Calibre digital, 150 mm MITUTOYO	122,57 €	3 años	1 semana	0,8 €
Micrómetro para medida de precisión de ejes. Ext. Micrometer 0-25mm std. RATCHET	57,94 €	3 años	1 semana	0,36 €
<b>TOTAL</b>				89 €

Tabla 2-5 Coste de amortización de equipos

Programa	Precio	Periodo amortización	Tiempo Proyecto	Amortización
Windows 7 Home Premium (Incluido en coste de PC)	-	3 años	10 semanas	-
Suite ofimática libre "Apache OpenOffice"	-	-	10 semanas	-
Editor IDE gratuito "MPLAB X" de Microchip	-	-	5,7 semanas	-
Software de diseño mecánico Solid Edge ST de SIEMENS edición universitaria (Licencia 1 año)	-	-	4,8 semanas	-
MS Adams Student Edition	-	-	4,8 semanas	-
<b>TOTAL</b>				0 €

Tabla 2-6 Coste de amortización de software

## 2.4. Total presupuesto de ejecución material

---

Coste de material Tabla 2-1	4.220,53 €
Coste de tornillería Tabla 2-2	48,24 €
Coste de portes Tabla 2-3	0,00 €
Coste de mano de obra Tabla 2-4	12.616,00 €
Coste de amortización de equipos Tabla 2-5	89 €
Coste de amortización de programas de trabajo Tabla 2-6	0,00 €
<b>TOTAL EJECUCIÓN MATERIAL</b>	<b>16.973,77 €</b>

Tabla 2-7 *Tabla total presupuesto ejecución material*

### **3. Gastos generales y beneficio industrial**

En este apartado se incluyen los gastos necesarios para disponer de instalaciones donde llevar a cabo los trabajos, amortizaciones, cargas, etc., más el Beneficio Industrial. Por estos conceptos se establece un recargo del 10% de gastos generales y un 15% de beneficio industrial sobre el presupuesto de Ejecución de Material.

25% del presupuesto total de ejecución de material

<u>Gastos generales y beneficio industrial</u>	<u>4.243,44 €</u>
--	-------------------

### **4. Coste de ejecución por contrata**

Se obtiene sumando el Coste de Ejecución Material, los Gastos Generales y el Beneficio Industrial.

<b>Coste de Ejecución material</b>	<b>16.972,61 €</b>
<b>Gastos generales y beneficio industrial</b>	<b>4.243,44 €</b>
<b>TOTAL EJECUCIÓN POR CONTRATA</b>	<b>21.216,05 €</b>

Tabla 4-1 *Tabla coste de ejecución por contrata*

### **5. Importe total de proyecto**

El importe total del proyecto es la suma del Coste de Ejecución por Contrata más el 21% de I.V.A.

<b>Importe de ejecución por contrata</b>	<b>21.216,05 €</b>
IVA 21%	4.455,37 €
<b>TOTAL CON IMPUESTOS</b>	<b>25.671,42 €</b>

Tabla 5-1 *Importe total proyecto*

# **MANUAL DE USUARIO**

# 1. Protocolo de comunicación I2C entre microcontroladores

---

Como ya se ha detallado las tareas de control del robot han sido repartidas entre los dos microcontroladores con los que cuenta la tarjeta de control. De tal forma que mientras en el maestro se centraliza todo el control de alto nivel, el esclavo ha sido supeditado para el control de los sistemas de manipulación de elementos de juego, bajo las órdenes del maestro.

Este diseño ha hecho necesaria la comunicación entre los dos microcontroladores y la creación de un protocolo común, que permita una comunicación segura.

Como bus de comunicación se ha elegido el bus i2c, que conecta el microcontrolador maestro con el esclavo, además de con la tarjeta de expansión de entradas-salidas que da soporte al mismo. Este bus nos va a permitir crear una comunicación maestro-esclavo según el rol de cada microcontrolador. Estableciendo una comunicación siempre de maestro a esclavo, con órdenes de lectura o escritura.

Esta fase del diseño se ha realizado de forma conjunta con el programador del microcontrolador maestro, definiendo las estructuras de datos que serán compartidas en la comunicación. Se ha creado un archivo llamado “i2c\_comands.h” que es compartido por los proyectos de ambos micro controladores.

En este archivo se han definido las estructuras de datos que pueden ser utilizadas. Siendo estas solo de dos tipos según se solicite una operación de lectura, o de escritura desde el maestro:

- <Operación de escritura>  
Solicitud de registro → I2C\_CMD\_GENERIC (definido en i2c\_comands.h)  
Se recibe la estructura: “i2c\_cmd\_slavedspic\_set\_mode”
  
- <Operación de lectura>  
Solicitud de registro → I2C\_REQ\_SLAVEDSPIC\_STATUS (def. en i2c\_comands.h)  
Se envía la estructura: “i2c\_slavedspic\_status”

## 1.1. Estructura i2c\_cmd\_slavedspic\_set\_mode

---

En esta estructura están codificadas todas las posibles órdenes, o comandos que el micro maestro puede solicitar al esclavo. Básicamente está formada por la variable “mode” donde se codifica el tipo de acción a realizar y por una unión de estructuras donde se codifican cada uno de los parámetros que definen cada acción.

En la programación del esclavo será creada una variable mainboard\_command de este tipo de estructura de datos donde se almacenará el comando a ejecutar. Siendo sobrescrita toda la estructura cada vez que se reciba un nuevo comando.

Esta es la lista de todos los comandos u órdenes que puede recibir el micro esclavo para el control de los sistemas de juego, entre paréntesis se indica su alias corto:

### 1.1.1. <Comandos de inicialización>

---

- I2C\_SLAVEDSPIC\_MODE\_INIT\_LEFT (INIT\_LEFT)  
Comando que inicializa todos los actuadores a la posición de comienzo del partido, jugando por el campo izquierdo.
- I2C\_SLAVEDSPIC\_MODE\_INIT\_RIGHT (INIT\_RIGHT)  
Comando que inicializa todos los actuadores a la posición de comienzo del partido, jugando por el campo derecho.
- I2C\_SLAVEDSPIC\_MODE\_POWER\_OFF (POWER\_OFF)  
Comando que detiene el movimiento de todos los actuadores. Detiene el carro horizontal, desactiva el torque de los servos Dynamixel, desactiva la pwm de los servos analógicos y por último pone la variable de estado en LISTO.

### 1.1.2. <Comandos para control de actuadores simples>

---

Son comandos de acción sobre un único actuador. Estos comandos han sido creados inicialmente solo para uso en depuración. Bloquean la ejecución del micro esclavo hasta que concluyen su acción.

- I2C\_SLAVEDSPIC\_MODE\_STANDS\_BLADE (STANDS\_BLADE)  
Comando de control de posición de ángulo de las paletas de stands.  
Estructura de parámetros asociada:
  - <i2c\_cmd\_slavedspic\_set\_mode.stands\_blade.type>  
Mediante type se selecciona la paleta sobre la que se aplicara el movimiento. Pudiendo ser la derecha o la izquierda.  
I2C\_STANDS\_BLADE\_TYPE\_LEFT  
I2C\_STANDS\_BLADE\_TYPE\_RIGHT
  - <i2c\_cmd\_slavedspic\_set\_mode.stands\_blade.mode>  
Mediante mode se selecciona la posición a la que se desplazara la paleta seleccionada más un offset desde esta posición, si este ha sido definido. Existe una excepción para el mode “SET\_ANGLE” que no corresponde a una posición, sino que produce un movimiento a la posición absoluta definida como offset.

- I2C\_STANDS\_BLADE\_MODE\_HIDE\_LEFT (Posición oculta en lado izquierdo+offset)
  - I2C\_STANDS\_BLADE\_MODE\_PUSH\_STAND\_LEFT (Posición de bloqueo de stand contra lado izquierdo+offset)
  - I2C\_STANDS\_BLADE\_MODE\_CENTER (Posición central+offset)
  - I2C\_STANDS\_BLADE\_MODE\_PUSH\_STAND\_RIGHT (Posición de bloqueo de stand contra lado derecho+offset)
  - I2C\_STANDS\_BLADE\_MODE\_HIDE\_RIGHT (Posición oculta en lado derecho+offset)
  - I2C\_STANDS\_BLADE\_MODE\_SET\_ANGLE (Posición absoluta como offset)
- <i2c\_cmd\_slavedspic\_set\_mode.stands\_blade.offset>  
Mediante “offset” se define una corrección en grados de la posición destino desde la seleccionada en “mode”. Con la excepción del mode “SET\_ANGLE”, en la que será tomada como una posición absoluta de destino. La posición central de la paleta corresponde al ángulo absoluto de 0 grados y el giro hacia el lado derecho a ángulos positivos.
- I2C\_SLAVEDSPIC\_MODE\_STANDS\_CLAMP (STANDS\_CLAMP)  
Comando de control de posición de ángulo de apertura de las pinzas de los elevadores de stands.  
Estructura de parámetros asociada:
  - <i2c\_cmd\_slavedspic\_set\_mode.stands\_clamp.type>  
Mediante type se selecciona el elevador sobre el que se aplicara el movimiento de la pinza. Pudiendo ser el derecho o el izquierdo.  
I2C\_STANDS\_CLAMP\_TYPE\_LEFT  
I2C\_STANDS\_CLAMP\_TYPE\_RIGHT
  - <i2c\_cmd\_slavedspic\_set\_mode.stands\_clamp.mode>  
Mediante mode se selecciona la posición de apertura más un offset desde esta posición, si este ha sido definido.  
I2C\_STANDS\_CLAMP\_MODE\_FULL\_OPEN (Posición de apertura máxima, utilizada para liberar las torres construidas+offset)  
I2C\_STANDS\_CLAMP\_MODE\_OPEN (Posición de pinza abierta+offset)  
I2C\_STANDS\_CLAMP\_MODE\_CLOSE (Posición de pinza cerrada+offset)
  - <i2c\_cmd\_slavedspic\_set\_mode.stands\_clamp.offset>  
Mediante “offset” se define una corrección en incrementos de servo de la posición destino desde la seleccionada en “mode”.
- I2C\_SLAVEDSPIC\_MODE\_STANDS\_ELEVATOR (STANDS\_ELEVATOR)  
Comando de control de posición de los elevadores de stands.

Estructura de parámetros asociada:

- <i2c\_cmd\_slavedspic\_set\_mode.stands\_elevator.type>  
Mediante type se selecciona el elevador sobre el que se aplicara el movimiento. Pudiendo ser el derecho o el izquierdo.  
I2C\_STANDS\_ELEVATOR\_TYPE\_LEFT  
I2C\_STANDS\_ELEVATOR\_TYPE\_RIGHT
- <i2c\_cmd\_slavedspic\_set\_mode.stands\_elevator.mode>  
Mediante mode se selecciona la posición a la que se desplazara el elevador seleccionado más un offset desde esta posición, si este ha sido definido.  
I2C\_STANDS\_ELEVATOR\_MODE\_UP (Posición superior o de almacenaje+offset)  
I2C\_STANDS\_ELEVATOR\_MODE\_DOWN (Posición inferior o de recogida+offset)
- <i2c\_cmd\_slavedspic\_set\_mode.stands\_elevator.offset>  
Mediante “offset” se define una corrección en incrementos de servo de la posición destino desde la seleccionada en “mode”.
- I2C\_SLAVEDSPIC\_MODE\_STANDS\_TOWER\_CLAMPS  
(STANDS\_TOWER\_CLAMPS)  
Comando de control de apertura de las puertas de las torres de stands.  
Estructura de parámetros asociada:
  - <i2c\_cmd\_slavedspic\_set\_mode.stands\_tower\_clamps.mode>  
Mediante mode se selecciona la posición de las dos puertas de las torres de stands más un offset, si este ha sido definido.  
I2C\_STANDS\_TOWER\_CLAMPS\_MODE\_UNLOCK\_LEFT (Puerta izquierda en posición de apertura+offset, la puerta derecha permanece cerrada)  
I2C\_STANDS\_TOWER\_CLAMPS\_MODE\_LOCK (Posición de cierre de ambas puertas+offset)  
I2C\_STANDS\_TOWER\_CLAMPS\_MODE\_UNLOCK\_RIGHT (Puerta derecha en posición de apertura+offset, la puerta izquierda permanece cerrada)
  - <i2c\_cmd\_slavedspic\_set\_mode.stands\_tower\_clamps.offset>  
Mediante “offset” se define una corrección en incrementos de servo de la posición destino desde la seleccionada en “mode”.
- I2C\_SLAVEDSPIC\_MODE\_CUP\_CLAMP\_POPCORN\_DOOR  
(CUP\_CLAMP\_POPCORN\_DOOR)

Comando de control de posición de ángulo de las pinzas del vaso trasero. La posición de apertura máxima produce indirectamente la apertura de las puertas de descarga de las palomitas.

Estructura de parámetros asociada:

- <i2c\_cmd\_slavedspic\_set\_mode.cup\_clamp\_popcorn\_door.type>  
Mediante type se selecciona la pinza o la puerta sobre la que se aplicara el movimiento. Pudiendo ser la derecha o la izquierda. (Siempre mirando desde fuera del robot)  
I2C\_CUP\_CLAMP\_POPCORN\_DOOR\_TYPE\_LEFT  
I2C\_CUP\_CLAMP\_POPCORN\_DOOR\_TYPE\_RIGHT
- <i2c\_cmd\_slavedspic\_set\_mode.cup\_clamp\_popcorn\_door.mode>  
Mediante mode se selecciona la posición a la que se desplazara la pinza seleccionada más un offset desde esta posición, si este ha sido definido.  
I2C\_CUP\_CLAMP\_MODE\_HIDE (Pinza en posición oculta+offset)  
I2C\_CUP\_CLAMP\_MODE\_LOCKED (Pinza en posición de bloqueo de vaso+offset)  
I2C\_CUP\_CLAMP\_MODE\_OPEN (Pinza en posición de coger-dejar vaso+offset)  
I2C\_POPCORN\_DOOR\_MODE\_CLOSE (Pinza en posición de coger-dejar vaso+offset)  
I2C\_POPCORN\_DOOR\_MODE\_OPEN (Pinza en posición de máxima apertura e indirectamente apertura de puerta de descarga de palomitas+offset)
- <i2c\_cmd\_slavedspic\_set\_mode.cup\_clamp\_popcorn\_door.offset>  
Mediante “offset” se define una corrección en incrementos de servo de la posición destino desde la seleccionada en “mode”.
- I2C\_SLAVEDSPIC\_MODE\_POPCORN\_TRAY (POPCORN\_TRAY)  
Comando de control de apertura de la bandeja de las palomitas.

Estructura de parámetros asociada:

- <i2c\_cmd\_slavedspic\_set\_mode.popcorn\_tray.mode>  
Mediante mode se selecciona la posición de la bandeja más un offset, si este ha sido definido.  
I2C\_POPCORN\_TRAY\_MODE\_OPEN (Posición puerta abierta+offset)  
I2C\_POPCORN\_TRAY\_MODE\_CLOSE (Posición puerta cerrada+offset)
- <i2c\_cmd\_slavedspic\_set\_mode.popcorn\_tray.offset>  
Mediante “offset” se define una corrección en incrementos de servo de la posición destino desde la seleccionada en “mode”.

- I2C\_SLAVEDESPIC\_MODE\_POPCORN\_RAMPS (POPCORN\_RAMPS)  
Comando de control de apertura de las rampas de las palomitas.  
Estructura de parámetros asociada:
  - <i2c\_cmd\_slavedspic\_set\_mode.popcorn\_ramps.mode>  
Mediante mode se selecciona la posición de las dos rampas de manera simétrica más un offset, si este ha sido definido.  
I2C\_POPCORN\_RAMPS\_MODE\_HIDE (Posición oculta de ambas rampas+offset)  
I2C\_POPCORN\_RAMPS\_MODE\_HARVEST (Posición de recolección de palomitas de ambas rampas+offset)  
I2C\_POPCORN\_RAMPS\_MODE\_OPEN (Posición de máxima apertura de ambas rampas+offset)
  - <i2c\_cmd\_slavedspic\_set\_mode.popcorn\_ramps.offset>  
Mediante “offset” se define una corrección en incrementos de servo de la posición destino desde la seleccionada en “mode”.
- I2C\_SLAVEDESPIC\_MODE\_CUP\_CLAMP\_FRONT (CUP\_CLAMP\_FRONT)  
Comando de control de posición de ángulo de apertura de la pinza del vaso delantero.  
Estructura de parámetros asociada:
  - <i2c\_cmd\_slavedspic\_set\_mode.cup\_clamp\_front.mode>  
Mediante mode se selecciona la posición a la que se desplazara la pinza más un offset desde esta posición, si este ha sido definido.  
I2C\_CUP\_CLAMP\_FRONT\_MODE\_OPEN (Posición pinza abierta+offset)  
I2C\_CUP\_CLAMP\_FRONT\_MODE\_CUP\_LOCKED (Posición pinza cerrada+offset)
  - <i2c\_cmd\_slavedspic\_set\_mode.cup\_clamp\_front.offset>  
Mediante “offset” se define una corrección en incrementos de servo de la posición destino desde la seleccionada en “mode”.
- I2C\_SLAVEDESPIC\_MODE\_CUP HOLDER\_FRONT  
(CUP HOLDER\_FRONT)  
Comando de control de posición de ángulo del brazo de la pinza del vaso delantero.  
Estructura de parámetros asociada:
  - <i2c\_cmd\_slavedspic\_set\_mode.cup\_holder\_front.mode>  
Mediante mode se selecciona la posición a la que se desplazara el brazo de la pinza más un offset desde esta posición, si este ha sido definido.

- I2C\_CUP HOLDER\_FRONT\_MODE\_CUP\_HOLD (Posición de descarga de palomitas+offset)
- I2C\_CUP HOLDER\_FRONT\_MODE\_READY (Posición de captura de vaso+offset)
- I2C\_CUP HOLDER\_FRONT\_MODE\_HIDE (Posición de brazo oculto+offset)
- <i2c\_cmd\_slavedspic\_set\_mode.cup\_holder\_front.offset>  
Mediante “offset” se define una corrección en incrementos de servo de la posición destino desde la seleccionada en “mode”.

### **1.1.3. <Comandos para control de actuadores múltiples>**

Son comandos de acción sobre sistemas complejos formados por más de un actuador.

- I2C\_SLAVEDSPIC\_MODE\_POPCORN\_SYSTEM (POPCORN\_SYSTEM)  
Comando de control sobre el sistema de la máquina de las palomitas.  
Estructura de parámetros asociada:
  - <i2c\_cmd\_slavedspic\_set\_mode.popcorn\_system.mode>  
Mediante mode se selecciona la operación a realizar.
  - I2C\_SLAVEDSPIC\_MODE\_PS\_IDLE  
Comando que devuelve la máquina de estados asociado al sistema a su estado inicial de libre.
  - I2C\_SLAVEDSPIC\_MODE\_PS\_CUP\_FRONT\_READY  
Comando que mueve el brazo hasta la posición de captura del vaso delantero y abre la pinza.
  - I2C\_SLAVEDSPIC\_MODE\_PS\_CUP\_FRONT\_CATCH\_AND\_DROP  
Comando que cierra la pinza para coger un vaso delantero y abate el brazo hasta la posición de vaciado de este.
  - I2C\_SLAVEDSPIC\_MODE\_PS\_CUP\_FRONT\_RELEASE  
Comando que baja el vaso desde la posición de vaciado y abre la pinza para liberar el vaso delantero.
  - I2C\_SLAVEDSPIC\_MODE\_PS\_CUP\_FRONT\_HIDE  
Comando que cierra la pinza y repliega el brazo hasta la posición oculta.
  - I2C\_SLAVEDSPIC\_MODE\_PS\_CUP\_REAR\_OPEN  
Comando que abre las pinzas traseras por orden hasta la posición de captura del vaso. Primero abre la pinza derecha.
  - I2C\_SLAVEDSPIC\_MODE\_PS\_CUP\_REAR\_CATCH  
Comando que cierra las pinzas traseras hasta la posición de bloqueo del vaso.
  - I2C\_SLAVEDSPIC\_MODE\_PS\_CUP\_REAR\_RELEASE  
Comando que libera el vaso, abriendo las pinzas hasta la posición de captura del vaso.
  - I2C\_SLAVEDSPIC\_MODE\_PS\_MACHINES\_READY

Comando que por orden abre la bandeja y las rampas de palomitas hasta posición de apertura máxima.

**I2C\_SLAVEDSPIC\_MODE\_PS\_MACHINES\_HARVEST**

Comando que mueve las rampas de palomitas hasta la posición de recolección.

**I2C\_SLAVEDSPIC\_MODE\_PS\_MACHINES\_END**

Comando que por orden cierra las rampas y la bandeja de palomitas.

**I2C\_SLAVEDSPIC\_MODE\_PS\_STOCK\_DROP**

Comando que abre las pinzas del vaso trasero hasta la posición máxima, provocando la apertura de las puertas de descarga de las palomitas.

**I2C\_SLAVEDSPIC\_MODE\_PS\_STOCK\_END**

Comando que cierra por orden las pinzas del vaso trasero hasta la posición de pinzas ocultas, provocando el cierre de las puertas de descarga de las palomitas. Primero se cierra la pinza izquierda.

- **I2C\_SLAVEDSPIC\_MODE\_STANDS\_SYSTEM (STANDS\_SYSTEM)**

Comando de control sobre las dos torres de almacenamiento de stands.

Estructura de parámetros asociada:

- <i2c\_cmd\_slavedspic\_set\_mode.stands\_system.side>

Mediante side se selecciona la torre a la que se dirige el comando. Pudiendo ser la derecha o la izquierda. (Siempre mirando desde fuera del robot)

**I2C\_SIDE\_LEFT**

**I2C\_SIDE\_RIGHT**

- <i2c\_cmd\_slavedspic\_set\_mode.stands\_system.mode>

Mediante mode se selecciona la operación a realizar.

**I2C\_SLAVEDSPIC\_MODE\_PS\_IDLE**

Comando que devuelve la máquina de estados asociado al sistema a su estado inicial de libre.

**I2C\_SLAVEDSPIC\_MODE\_SS\_HIDE\_TOWER**

Comando que cierra la puerta de la torre.

**I2C\_SLAVEDSPIC\_MODE\_SS\_HARVEST\_STAND\_DO**

Comando que captura un stand y lo almacena.

**I2C\_SLAVEDSPIC\_MODE\_SS\_HARVEST\_STAND\_READY**

Comando que prepara todos los sistemas de la torre para capturar un stand.

**I2C\_SLAVEDSPIC\_MODE\_SS\_BUILD\_SPOTLIGHT**

Comando que construye una torre hasta de 8 stands.

**I2C\_SLAVEDSPIC\_MODE\_SS\_RELEASE\_SPOTLIGHT**

Comando que libera la torre.

- <i2c\_cmd\_slavedspic\_set\_mode.stands\_system.blade\_angle>

Mediante blade\_angle se define el ángulo inicial de apertura de las paletas para los comandos de captura de stands. Ya que dependiendo de

la posición del stand en el campo, la paleta tiene que estar posicionada desviada de la posición central para facilitar su captura. El valor ha de ser codificado en grados, correspondiendo la posición central a 0 grados y el giro hacia el lado derecho a ángulos positivos.

#### **1.1.4. <Comandos para ajuste de variables de estado>**

---

Este comando ajusta las variables de estado a los valores definidos en la estructura.

- I2C\_SLAVEDESPIC\_MODE\_SET\_INFOS (SET\_INFOS)
  - <i2c\_cmd\_slavedspic\_set\_mode.set\_infos.cup\_front\_catched>  
Mediante cup\_front\_catched se modifica la variable de estado de vaso delantero atrapado.
  - <i2c\_cmd\_slavedspic\_set\_mode.set\_infos.cup\_rear\_catched>  
Mediante cup\_rear\_catched se modifica la variable de estado de vaso trasero atrapado.
  - <i2c\_cmd\_slavedspic\_set\_mode.set\_infos.machine\_popcorns\_catched>  
Mediante machine\_popcorns\_catched se modifica la variable de estado de máquinas de palomitas recolectadas.
  - <i2c\_cmd\_slavedspic\_set\_mode.set\_infos.stored\_stands\_l>  
Mediante stored\_stands\_l se modifica la variable de estado de número de stands almacenados en lado izquierdo.
  - <i2c\_cmd\_slavedspic\_set\_mode.set\_infos.stored\_stands\_r>  
Mediante stored\_stands\_r se modifica la variable de estado de número de stands almacenados en lado derecho.

## **1.2. Estructura I2C\_REQ\_SLAVEDESPIC\_STATUS**

---

En esta estructura están codificadas todas las variables de estado que el micro maestro necesita monitorizar del esclavo. Cuando el esclavo reciba por i2c una solicitud para la lectura del registro “I2C\_REQ\_SLAVEDESPIC\_STATUS” se actualizarán todas las variables de la estructura y será enviada.

El micro maestro va a crear una tarea periódica para solicitar la lectura de esta estructura.

La estructura está formada por los siguientes variables:

- <i2c\_slavedspic\_status.status>

En la variable “status” se almacena el estado global de ejecución del esclavo. En esta variable de 8 bits cada uno de ellos corresponde a uno de los estados codificados, siendo “1” activo y “0” desactivado. Esto permite que puedan estar activos varios estados no excluyentes de manera simultánea.

### I2C\_SLAVEDESPIC\_STATUS\_BLOCKED

Un actuador no ha alcanzado la posición de destino, y se ha abortado la ejecución del comando programado. En el control de todos los servos Dynamixel, no se da por concluido el movimiento hasta que se ha alcanzado una posición cercana a la esperada y transcurrido el triple del tiempo calculado para el movimiento se considera el movimiento bloqueado, pasando al estado correspondiente.

### I2C\_SLAVEDESPIC\_STATUS\_ERROR

Ha ocurrido un error no esperado, y que no debería darse nunca.

### I2C\_SLAVEDESPIC\_STATUS\_READY

Estado de espera de nuevo comando.

### I2C\_SLAVEDESPIC\_STATUS\_BUSY

Estado de esclavo ocupado con comando en ejecución.

### I2C\_SLAVEDESPIC\_STATUS\_WAITING

Este estado solo se da en dos situaciones. Despues de ejecutar el comando “I2C\_SLAVEDESPIC\_MODE\_SS\_HARVEST\_STAND\_READY”, una vez que se ha preparado uno de los sistemas de almacenaje de stands para una captura y se está esperando a que sea capturado con el movimiento de avance del robot. Y una vez que ha entrado el stand y dada la orden de cerrar la paleta se está esperando a que esta alcance su posición de destino.

### I2C\_SLAVEDESPIC\_STATUS\_STORING

Este estado se da mientras se están almacenando los stands.

### I2C\_SLAVEDESPIC\_STATUS\_DONE

Este es un estado previo a READY que permite optimizar el código enlazando comandos para construir tareas más complejas. Permite saltar de un modo a otro automáticamente una vez finalizado el primero.

### I2C\_SLAVEDESPIC\_STATUS\_RESERVED

Estado no utilizado.

- <i2c\_slavedspic\_status.popcorn\_system.

Es una estructura donde se almacenan todas las variables de estado del sistema de recolección de palomitas.

**Mode>**

Estado de máquina de estado del sistema.

**Status>**

En la variable “status” se almacena el estado particular de este sistema. En esta variable de 8 bits cada uno de ellos corresponde a uno de los estados codificados, siendo “1” activo y “0” desactivo. Esto permite que puedan estar activos varios estados no excluyentes de manera simultánea.

**cup\_front\_catched>**

Esta variable tendrá valor positivo cuando el vaso delantero está capturado.

**cup\_rear\_catched>**

Esta variable tendrá valor positivo cuando el vaso trasero está capturado.

**machine\_popcorns\_catched>**

Esta variable tendrá valor positivo cuando hemos recolectado la máquina de palomitas y las tenemos almacenadas en el depósito.

**<i2c\_slavedspic\_status.stands\_system[2].**

Es un array de dos estructuras donde se almacenan todas las variables de estado de los dos sistemas de almacenaje de stands.

**Mode>**

Estado de máquina de estado del sistema.

**status>**

En la variable “status” se almacena el estado particular de este sistema. En esta variable de 8 bits cada uno de ellos corresponde a uno de los estados codificados, siendo “1” activo y “0” desactivo. Esto permite que puedan estar activos varios estados no excluyentes de manera simultánea.

**stored\_stands>**

Esta variable indica el número de stands que portamos en el almacén correspondiente.

# **ANEXOS**



# 1. Librerías AVERSIVE

---

Para la creación del programa se han utilizado como base las librerías Aversive. Se trata de unas librerías con licencia GPL creadas por el equipo de robótica de competición “Microb Technology”, que participó en la competición de Eurobot desde el año 2003 hasta 2010. Estas librerías fueron desarrolladas en lenguaje C para los microcontroladores AVR de Atmel y portadas posteriormente por Javier Baliñas Santos para los dsPIC de Microchip. Siendo utilizadas desde entonces por el equipo de robótica de competición “Eurobotics Engineering”.

Se trata de unas librerías altamente estructuradas y potentes, que abarcan desde el más bajo nivel hasta soporte para el control de sistemas de alto nivel, como puede ser la auto evitación de obstáculos en un sistema de control de trayectoria por odometría. En su contra el mayor inconveniente que presentan es la falta de documentación de soporte y la necesidad de análisis del código para su comprensión.

Como fase previa a la programación desarrollada se ha analizado y explicado el funcionamiento de aquellas funciones y módulos pertenecientes a las librerías Aversive que se han utilizado.

## 1.1. Organización de módulos

---

Los principales módulos en los que se encuentra dividida la librería Aversive son los siguientes:

1. Base: módulos útiles de funcionalidades varias:
  - base/cirbuf: implementación de buffer circulares.
  - base/scheduler: programación de eventos periódicos y de ejecución única con niveles de prioridad.
  - base/time: generador de tiempo de reloj.
  - base/math/vect2: Este módulo proporciona funciones y estructuras de datos para manejar vectores de 2 dimensiones tanto en coordenadas cartesianas como polares. No ha sido utilizado.
  - base/math/fixed\_pint: No ha sido utilizado.
2. Comm: módulos relativos a protocolos de comunicación o hardware de comunicación del dsPIC:
  - comm/uart
  - comm/i2c
  - comm/mf2\_client
  - comm/mf2\_server
  - comm/spi
3. Crypto: algunos módulos utilizados para hacer operaciones criptográficas:
  - crypto/aes
  - crypto/md5
  - crypto/rc4

4. Debug: módulos que facilitan la depuración del código:
  - debug/diagnostic
  - debug/error
5. Devices: contiene módulos que están relacionados con un dispositivo que no es parte de un dsPIC (LCD, motores,...). Por lo general, un módulo se relaciona con un dispositivo electrónico específico que se describe en la fuente de módulo:
  - devices/brushless\_motors/brushless\_3phase\_digital\_hall: No ha sido utilizado.
  - devices/brushless\_motors/brushless\_3phase\_digital\_hall\_double: No ha sido utilizado.
  - devices/control\_system/control\_system\_manager: Módulo de programación de bucles de control
  - devices/control\_system/filters/biquad: No ha sido utilizado.
  - devices/control\_system/filters/pid: Filtro genérico PID para control de bucle.
  - devices/control\_system/filters/quadramp: Filtro de consigna de generación de rampas de arranque y parada, con definición de límite de velocidad asimétrica y aceleración constante asimétrica.
  - devices/control\_system/filters/quadramp\_derivate: No ha sido utilizado.
  - devices/control\_system/filters/ramp: No ha sido utilizado.
  - devices/encoders/encoders\_eirbot: Módulo para el control de encóder.
  - devices/encoders/encoders\_microb
  - devices/ihm/lcd: No ha sido utilizado.
  - devices/robot
  - devices/robot/position\_manager: No ha sido utilizado.
  - devices/robot/robot\_system: No ha sido utilizado.
  - devices/robot/trajectory\_manager: No ha sido utilizado.
  - devices/servo/ax12: Módulo para el control de los servos AX12 de Dynamixel.
6. Encoding: operaciones para buffers de codificación:
  - encoding/base64
  - encoding/hamming
7. Hardware: contiene módulos que hacen uso de hardware específico del dsPIC (ADC, PWM):
  - hardware/adc
  - hardware/pwm
  - hardware/timer
8. Ihm: Interfaz hombre-máquina.
  - ihm/menu

Para simplificar el análisis y facilitar la comprensión tan solo se van a comentar aquellos módulos que han sido utilizadas en el código creado.

## **1.2. base/scheduler**

---

Este módulo permite programar (planificar) la ejecución de funciones definidas por el usuario de manera periódica o de ejecución única, con posibilidad de configuración del intervalo entre cada llamada a la función y el nivel de prioridad. Durante la ejecución de la función las interrupciones son enmascaradas por lo que será importante reducir al máximo el tamaño de las tareas introducidas para evitar posibles errores.

La temporización entre eventos y la base de tiempo están controladas por la interrupción periódica del timer 1. El intervalo entre cada llamada a una función programada será múltiplo del periodo de dicha interrupción y se indicara en número de periodos. Esta interrupción ejecuta la función “scheduler\_interrupt(void)” que es la encargada del control de eventos. Si varios eventos coinciden en la misma interrupción estos serán atendidos según su nivel de prioridad en sucesivas interrupciones. Los niveles de prioridad configurables son de 0 a 255, correspondiendo 255 a la mayor prioridad en ejecución.

Asociado a este módulo existe un archivo de configuración llamado “scheduler\_config.h”. En este archivo de cabecera se han agrupado todas las configuraciones del módulo. Las más importantes son:

- **SCHEDULER\_UNIT**  
Define el número de microsegundos entre cada interrupción del timer 1 y por lo tanto la base de tiempos. Como norma el parámetro “period” de las funciones del módulo es definido como “intervalo entre cada llamada a la función (microsegundos)"/SCHEDULER\_UNIT (microsegundos/periodo scheduler). Para realizar la definición en este archivo de cabecera tendrá que ser definido “CONFIG\_MODULE\_SCHEDULER\_MANUAL”, en caso contrario la definición se realizará de forma automática en el archivo de cabecera “scheduler.h”.
- **SCHEDULER\_NB\_MAX\_EVENT**: define el máximo número de eventos que pueden ser creados. Por defecto este es de 8.
- **SCHEDULER\_TIMER\_NUM**: define el número del timer utilizado por el scheduler, pero no es de utilidad para los dsPIC, ya que siempre es utilizado en timer 1.

Para simplificar el análisis y facilitar la comprensión tan solo se van a comentar aquellas funciones que han sido utilizadas en el código creado o son utilizadas en los módulos que han sido adaptados para el control de la electrónica dedicada a los sistemas de manipulación de elementos de juego.

- **void scheduler\_init(void)**  
Esta función inicializa el módulo, borra la estructura de definición de eventos y configura el timer 1 (caso del dsPIC) si ha sido definido “CONFIG\_MODULE\_SCHEDULER\_TIMER1” con la frecuencia definida por “SCHEDULER\_UNIT” y habilita su interrupción iniciando el scheduler. En el caso de que “CONFIG\_MODULE\_SCHEDULER\_TIMER1” no haya sido definido previamente la configuración del timer1 tendrá que realizarse manualmente en el código de usuario.
- **void scheduler\_dump\_events(void)**  
Esta función imprime por la salida estándar con formato todos los eventos configurados para el scheduler y almacenados en el array de estructuras “g\_tab\_event[SCHEDULER\_NB\_MAX\_EVENT]”.

- static inline int8\_t **scheduler\_add\_periodical\_event\_priority**(void (\*f)(void \*), void \* data, uint16\_t period, uint8\_t priority)

Esta función añade al scheduler un evento, consistente en la ejecución de una función, que se ejecutara en el intervalo de tiempo definido de forma periódica y según un nivel de prioridad como ya se ha explicado.

*Argumentos:*

- Puntero a la función programada.
- Puntero a los parámetros de la función programada.
- Intervalo entre cada llamada a la función programada (nº de interrupciones de scheduler).
- Prioridad de ejecución de la función programada (0-255).

*Retorna:* el id asignado al evento y “-1” en caso de error.

- uint8\_t **scheduler\_disable\_save**(void)

Esta función deshabilita la ejecución de eventos del scheduler. La ejecución de los eventos es retrasada hasta que vuelva a ser habilitada.

*Retorna:* la prioridad del evento en ejecución. Es necesario almacenar este valor para iniciar la ejecución del scheduler con la función “scheduler\_enable\_restore” que lo recibirá como parámetro. Estas dos funciones son utilizadas para detener momentáneamente el scheduler en secciones de ejecución crítica, como ocurre en el módulo de comunicación por i2c de los servos Dynamixel AX12.

- void **scheduler\_enable\_restore**(uint8\_t old\_prio)

Esta función habilita la ejecución de eventos del scheduler tras la deshabilitación por la función “scheduler\_disable\_save”.

*Argumentos:*

- Nivel de prioridad del evento en ejecución en el momento de la deshabilitación del scheduler.

---

### 1.3. base/time

---

Este módulo permite obtener un tiempo de reloj en segundos y microsegundos, e inicializarlo o ajustarlo bajo demanda. Está diseñado para que resulte muy fácil de utilizar pero en su contra no tiene excesiva precisión. Para obtener la máxima precisión es necesario configurarlo atendiendo a lo siguiente.

Asociado a este módulo existe un archivo de configuración llamado “time\_config.h”. En este archivo de cabecera tan solo se encuentra la definición de “TIME\_PRECISION” que corresponde a la base de tiempos en microsegundos del módulo. Este módulo utiliza el scheduler para generar un evento periódico con periodo definido por (int)((float)(TIME\_PRECISION)) / SCEDULER\_UNIT\_FLOAT ), siendo “SCEDULER\_UNIT\_FLOAT” igual que “SCEDULER\_UNIT” que ya se ha definido en el módulo scheduler pero con formato float. Teniendo en cuenta la conversión “cast” a entero que es necesario hacer para calcular el número de cuentas de interrupciones del scheduler que equivaldrán al “TIME\_PRECISION”, habrá que tener en cuenta la perdida de precisión y por

lo tanto para minimizar esta, definir un “TIME\_PRECISION” múltiplo de “SCHEDULER\_UNIT”.

El cálculo del tiempo transcurrido cada vez que se genere el evento con la periodicidad programada en el scheduler se realizara mediante la siguiente fórmula:

```
us_actual(us) = us_anterior_evento(us) +
((int)(TIME_PRECISION(us) / SCHEDULER_UNIT)) * SCHEDULER_UNIT
```

Para simplificar el análisis y facilitar la comprensión tan solo se van a comentar aquellas funciones que han sido utilizadas en el código creado o son utilizadas en los módulos que han sido adaptados para el control de la electrónica dedicada a los sistemas de manipulación de elementos de juego.

- void **time\_init**(uint8\_t priority)

Esta función inicializa el módulo, creando un evento periódico en el scheduler con un intervalo de ejecución definido por “TIME\_PRECISION” y “SCHEDULER\_UNIT” según se ha explicado y un nivel de prioridad “priority”. También inicializa la cuenta de segundos y microsegundos.

*Argumento:*

- Prioridad de ejecución de la función programada en el scheduler.

- microseconds **time\_get\_us**(void)

Esta función devuelve el tiempo en microsegundos desde la última inicialización o reset. Para obtener el tiempo total, transcurrido más de un segundo, será necesario aplicar la siguiente formula:

$$\text{microsegundos\_totales} = \text{microsegundos} + (\text{segundos} \times 1000)$$

Siendo los segundos obtenidos con la función “time\_get\_s()”.

*Retorna:* Los microsegundos codificados en una variable entera con signo de 32 bits con la particularidad explicada.

- seconds **time\_get\_s**(void)

Esta función devuelve el tiempo en segundos desde la última inicialización o reset.

*Retorna:* Los segundos codificados en una variable entera con signo de 32 bits.

- microseconds **time\_get\_us2**(void)

Esta función devuelve el tiempo en microsegundos con desbordamiento natural de la variable contenedora.

*Retorna:* Los microsegundos codificados en una variable entera con signo de 32 bits con la particularidad explicada.

- `time_h time_get_time(void)`

Esta función devuelve la estructura de tiempo, que contiene los microsegundos (us) y segundos (s). Con estos datos es posible calcular el tiempo en microsegundos transcurrido desde la última inicialización o reset.

*Retorna:* La estructura de tiempo “`time_h`”.

## 1.4. comm/uart

---

Está librería da soporte hasta a cuatro módulos UART, con unos buffer FIFO de recepción y transmisión asociado a cada uno.

Al igual que en otros módulos existe un archivo de configuración llamado “`uart_config.h`”. En este archivo de cabecera se encuentra la definición de cada UART y se habilita la compilación de su código asociado, mediante la definición del correspondiente “`UARTX_COMPILE`”. En el caso del dsPIC `UART0_COMPILE` habilita la uart 1 y `UART2_COMPILE` la uart 2.

La llamada a la función “`uart_init()`” que no recibe parámetros, configurará automáticamente las uart habilitadas para la compilación, con los ajustes codificados en este archivo. Entre estos datos además de los de definición del protocolo serie asíncrono utilizado, está el tamaño de los buffer asociados.

Para simplificar el análisis y facilitar la comprensión tan solo se van a comentar aquellas funciones que han sido utilizadas en el código creado o son utilizadas en los módulos que han sido adaptados para el control de la electrónica dedicada a los sistemas de manipulación de elementos de juego.

- `void uart_register_rx_event(uint8_t num, void (*f)(char))`

Esta función registra otras funciones que serán ejecutadas cada vez que se reciba un byte en la uart definida.

*Argumentos:*

- Número de la uart sobre la que se registra la función.
- Puntero a la función registrada. Esta función recibirá como parámetro el byte recibido en la uart.

- `void uart_init(void)`

Inicialización de módulo. Esta función pone los registros asociados a cada uart (con compilación habilitada) en el estado correcto para empezar a utilizarla. Utilizando el archivo de configuración “`uart_config.h`”.

Utilizar esta función equivale a llamar a “`uartX_setconf(NULL)`” por cada uart configurada.

- `int uart_send(uint8_t num, char c)`

Esta función envía un byte a través de la uart “`num`”. El byte es almacenado en el buffer FIFO de transmisión y si este estuviera lleno, esta función espera hasta que la uart termine de enviar el siguiente byte.

*Argumentos:*

- Número de la uart.
- Byte a enviar.

*Retorna:* El byte enviado.

- int **uart\_recv\_nowait**(uint8\_t num)

Esta función retorna el próximo byte del buffer FIFO de recepción asociado a la uart “num”. Si no hay nada para leer la función no espera y retorna -1.

*Argumento:*

- Número de la uart.

*Retorna:* Siguiente byte recibido o -1 si no hay nada para leer.

## 1.5. comm/i2c\_slave

---

La librería de i2c utilizada en el microcontrolador esclavo es tan solo una versión ligera de la original y solo da soporte para la gestión de un módulo i2c como esclavo.

En los microcontroladores dsPIC el software de usuario es el responsable de la gestión y construcción de los mensajes con el protocolo i2c correcto. En esta librería esta tarea se gestiona desde la función de atención a la interrupción de eventos de esclavo “\_SI2C1Interrupt()”. Desde esta función según la operación solicitada desde el maestro sea de lectura o escritura se procede así:

Operaciones de lectura:

<Secuencia de mensaje>

1. <Recepción> Secuencia de inicio.
2. <Recepción> Dirección/Solicitud escritura > Se almacena la dirección temporalmente.
3. <Recepción> Número de registro a leer.
  - El número del registro es almacenado en la primera posición del array “rx\_data[0]”.
  - Se ejecuta la función definida por el usuario de la librería y apuntada por “read\_event”, en la que se deberán cargar todos los datos del registro solicitado en el array “tx\_data[]”, para que a continuación según sean solicitados datos estos sean enviados desde “\_SI2C1Interrupt()”.
4. <Recepción> Secuencia de inicio.
5. <Recepción> Dirección/Solicitud lectura > Se almacena la dirección temporalmente.
  - Se envía el primer byte. Correspondiente a la primera posición del array “tx\_data[0]”.
6. <Recepción> Solicitud de nueva lectura.
  - Se envía el siguiente byte y así mientras se soliciten nuevos datos.

Operaciones de escritura:

<Secuencia de mensaje>

1. <Recepción> Secuencia de inicio.
2. <Recepción> Dirección/Solicitud escritura > Se almacena la dirección temporalmente.
3. <Recepción> Número de registro a escribir.
  - El número del registro es almacenado en la primera posición del array “rx\_data[0]”.
  - Se ejecuta la función definida por el usuario de la librería y apuntada por “write\_event”, en la que se deberá comprobar que el tamaño del dato recibido corresponde al esperado según el tipo de registro enviado. Esta función recibe el registro a escribir, la dirección del array que contiene el dato recibido y el tamaño del dato recibido hasta ese momento. En esta primera llamada a la función todavía no se habrá recibido ningún dato.
4. <Recepción> Byte de datos.
  - El byte es almacenado en la siguiente posición del array “rx\_data[]”.
  - Se ejecuta la función definida por el usuario de la librería y apuntada por “write\_event”, en la que se deberá comprobar que el tamaño del dato recibido corresponde al esperado según el tipo de registro enviado. Esta función recibe el registro a escribir, la dirección del array que contiene el dato recibido y el tamaño del dato recibido hasta ese momento.

Para simplificar el análisis y facilitar la comprensión tan solo se van a comentar aquellas funciones que han sido utilizadas en el código creado o son utilizadas en los módulos que han sido adaptados para el control de la electrónica dedicada a los sistemas de manipulación de elementos de juego.

- **void i2c\_init(uint8\_t addr)**

Inicialización de módulo. Esta función pone los registros asociados al módulo i2c en el estado correcto para empezar a utilizarlo. Este módulo no dispone de archivo de configuración.

*Argumento:*

- Número identificador del nodo en el bus i2c.

- **void i2c\_register\_write\_event(void (\*event)(uint8\_t, uint8\_t \*, uint16\_t))**

Mediante esta función se asigna al módulo i2c la función que atenderá las operaciones de escritura. Siendo invocada cada vez que sea recibido un byte en una operación de escritura.

La función creada tendrá que tener el formato definido, recibiendo el número del registro a escribir, un puntero al comienzo de los datos recibidos y el número de bytes recibidos.

*Argumento:*

- Puntero a la función a registrar.

- void **i2c\_register\_read\_event**(void (\*event)(uint8\_t, uint8\_t \*))

Mediante esta función se asigna al módulo i2c la función que atenderá las operaciones de lectura. Siendo invocada solamente cuando se reciba el primer byte de datos en una operación de escritura y que corresponderá al número del registro que es solicitado para lectura, según la secuencia de lectura explicada en el resumen del módulo más arriba. El cometido de la función creada será preparar los datos del registro solicitado introduciéndolos en el array de transmisión “tx\_data[]”.

La función creada tendrá que tener el formato definido, recibiendo el número del registro a leer y un puntero al array donde serán almacenados los datos para su transmisión.

*Argumento:*

- Puntero a la función a registrar.

---

## 1.6. debug/error

---

Este módulo implementa un sistema para el registro de errores y la ejecución de acciones definidas por el usuario según el tipo de error.

Crea una estructura de error, con los siguientes datos:

- Número de error.
- Nivel de error.
- Texto de definición de error.
- Fichero en el que se ha producido el error.
- Línea en la que se ha producido el error.

Define 5 niveles de error, que serán los codificables como “Nivel de error”.

- ERROR\_SEVERITY\_EMERG 0
- ERROR\_SEVERITY\_ERROR 1
- ERROR\_SEVERITY\_WARNING 2
- ERROR\_SEVERITY\_NOTICE 3
- ERROR\_SEVERITY\_DEBUG 4

Crea una función por cada tipo de nivel de error para almacenar en un array de punteros a funciones de atención error, la dirección de la función que será ejecutada cuando se registre un error. En el prototipo de estas funciones de registro ya se define que las de atención recibirán como parámetro una estructura de error.

Por último se definen unas macros por cada tipo de nivel de error, con este formato “EMERG(num, text, ...)”. Que serán las utilizadas para registrar los errores. Estas macros llenan todas las variables de una estructura de error y la pasan como parámetro a la función de atención de nivel de error definida y registrada por el usuario de la librería.

Para simplificar el análisis y facilitar la comprensión tan solo se van a comentar aquellas funciones que han sido utilizadas en el código creado o son utilizadas en los módulos

que han sido adaptados para el control de la electrónica dedicada a los sistemas de manipulación de elementos de juego.

- `void error_register_emerg(void (*f)(struct error *, ...))`

Esta función registra la función definida por el usuario y que será ejecutada cuando se utilice la macro de nivel de error de emergencia “EMERG(num, text, ...)”.

*Argumento:*

- Puntero a estructura de error.

Existen otras cuatro funciones con una estructura idéntica y que realizan la misma función para los otros niveles de error.

## **1.7. devices/control\_system/control\_system\_manager**

Este módulo tiene como objetivo proporcionar una arquitectura muy genérica y portátil para la programación de bucles de control. El sistema de control es portable, ya que está independizado de las funciones de entrada salida. Y es posible definir tantos bucles de control como sean necesarios.

Su estructura básica es muy sencilla, se trata de un bucle de retroalimentación con tres filtros que son dinámicamente reconfigurables y opcionales. El bloque de entrada al proceso corresponde al módulo que aplica la variable de salida al proceso. Y el bloque de salida corresponde al módulo de captura de la señal de salida del proceso y conversión al tipo de variable de la consigna.

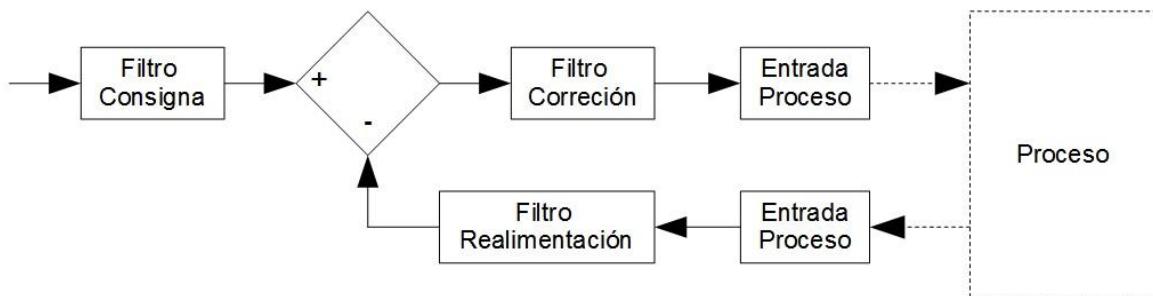


Diagrama 1-1 *Diagrama bucle de control de módulo "control\_system\_manager"*

Toda esta estructura está implementada de una manera muy sencilla en la función de control de bucles. El prototipo de esta función es la siguiente:

- `int32_t cs_do_process(struct cs* cs, int32_t consign)`

Esta función recibe como parámetros la estructura de control de bucle y el valor de consigna, y devuelve el valor de salida del proceso. Como norma general el valor de salida es aplicado al sistema en el bloque de entrada de proceso.

En la estructura de control de bucle se encuentra toda la información que necesita esta función para operar, con la única excepción de las estructuras de

datos de los filtros y los bloques de entrada-salida que hayan sido definidas, pero que si se encuentran direccionadas desde la estructura de control de bucle a través de unos punteros a dichas estructuras.

Sintetizando, su contenido sería:

- 5 Punteros a cada una de las funciones que ejecutaran cada uno de los 5 bloques del sistema. Estas funciones reciben como parámetros un puntero a la estructura de datos del filtro y el valor de entrada, y devuelven el valor de salida.
- 5 Punteros a las estructuras de datos de cada uno de los 5 bloques del sistema.
- 5 Variables correspondientes al valor de consigna, al valor de consigna filtrado, al valor de salida de proceso (previo al filtro de realimentación), al valor realimentado filtrado y al valor de error (previo al filtro de corrección).

Por lo tanto esta función se limita a ejecutar en un orden determinado las funciones de cada uno de los filtros asignados a cada bloque y a almacenar el resultado del proceso en las 5 variables intermedias de salida de cada bloque. En último lugar en la función del bloque de entrada del proceso, aplica el valor de salida resultante.

El control de la ejecución de esta función recae sobre el usuario que tiene que ejecutarla periódicamente.

Tanto los filtros como los módulos de entrada y salida son la unidad mínima del sistema y se han creado con una estructura común. Esto permite que durante la inicialización de un bucle de control se pueda asignar cualquier filtro a los distintos bloques del sistema.

Cada filtro y módulo de entrada salida dispone de los siguientes elementos:

- Estructura de datos, adaptada a las necesitadas de cada filtro, dentro de ella se almacenan todas las variables internas del filtro.
- Función de inicialización de los valores de la estructura. Se le pasa como parámetro un puntero al tipo de estructura del filtro.
- Función de ejecución del filtro que realiza una iteración de control del filtro. Recibe como parámetros un puntero a la estructura del filtro y el valor de entrada, y devuelve el valor de salida del filtro. Un ejemplo del prototipo de esta función es s32 example\_do\_filter (struct example\_filter \* p, S32 in).

Las librerías Aversive disponen de 5 tipos de filtros ya programados. Estos son PID, Rampa, Quadramp, Quadramp Derivate y Biquad.

Para simplificar el análisis y facilitar la comprensión tan solo se van a comentar aquellas funciones que han sido utilizadas en el código creado o son utilizadas

en los módulos que han sido adaptados para el control de la electrónica dedicada a los sistemas de manipulación de elementos de juego.

- **void cs\_init(struct cs\* cs)**

Esta función borra todas las variables de la estructura de control de bucle apuntada por `cs`.

*Argumentos:*

- Puntero a estructura de control de bucle

- **void cs\_set\_consign\_filter(struct cs\* cs, int32\_t (\*consign\_filter)(void\*, int32\_t), void\* consign\_filter\_params)**

Esta función guarda en la estructura de control de bucle, apuntada por “`cs`”, la dirección de la función y la estructura del filtro que se ejecutara como filtro de consigna.

*Argumentos:*

- Puntero a estructura de control de bucle.
- Puntero a función de filtro.
- Puntero a estructura de filtro.

- **void cs\_set\_correct\_filter(struct cs\* cs, int32\_t (\*correct\_filter)(void\*, int32\_t), void\* correct\_filter\_params)**

Esta función guarda en la estructura de control de bucle, apuntada por “`cs`”, la dirección de la función y la estructura del filtro que se ejecutara como filtro de corrección.

*Argumentos:*

- Puntero a estructura de control de bucle.
- Puntero a función de filtro.
- Puntero a estructura de filtro.

- **void cs\_set\_feedback\_filter(struct cs\* cs, int32\_t (\*feedback\_filter)(void\*, int32\_t), void\* feedback\_filter\_params)**

Esta función guarda en la estructura de control de bucle, apuntada por “`cs`”, la dirección de la función y la estructura del filtro que se ejecutara como filtro de realimentación.

*Argumentos:*

- Puntero a estructura de control de bucle.
- Puntero a función de filtro.
- Puntero a estructura de filtro.

- **void cs\_set\_process\_in(struct cs\* cs, void (\*process\_in)(void\*, int32\_t), void\* process\_in\_params)**

Esta función guarda en la estructura de control de bucle, apuntada por “`cs`”, la dirección de la función y la estructura asociada al bloque de control de entrada de proceso.

*Argumentos:*

- Puntero a estructura de control de bucle.

- Puntero a función de filtro.
- Puntero a estructura de filtro.
- void **cs\_set\_process\_out**(struct cs\* cs, int32\_t (\*process\_out)(void\*), void\* process\_out\_params)
 

Esta función guarda en la estructura de control de bucle, apuntada por “cs”, la dirección de la función y la estructura asociada al bloque de control de salida de proceso.

*Argumentos:*

- Puntero a estructura de control de bucle.
- Puntero a función de filtro.
- Puntero a estructura de filtro.

El acceso a las distintas variables de la estructura de datos de control de bucle, ha de hacerse a través de las siguientes funciones. Estas funciones deshabilitan las interrupciones antes de acceder a las variables para evitar la posible corrupción de datos.

- int32\_t **cs\_get\_out**(struct cs\* cs)

*Argumentos:*

- Puntero a estructura de control de bucle.

*Retorna:* Valor de salida del filtro de corrección.

- int32\_t **cs\_get\_error**(struct cs\* cs)

*Argumentos:*

- Puntero a estructura de control de bucle.

*Retorna:* Valor de error (previo al filtro de corrección).

- int32\_t **cs\_get\_consign**(struct cs\* cs)

*Argumentos:*

- Puntero a estructura de control de bucle.

*Retorna:* Valor de consigna.

- int32\_t **cs\_get\_filtered\_consign**(struct cs\* cs)

*Argumentos:*

- Puntero a estructura de control de bucle.

*Retorna:* Valor de consigna filtrado.

- int32\_t **cs\_get\_filtered\_feedback**(struct cs\* cs)

*Argumentos:*

- Puntero a estructura de control de bucle.

*Retorna:* Valor realimentado filtrado.

- void **cs\_set\_consign**(struct cs\* cs, int32\_t v)

*Argumentos:*

- Puntero a estructura de control de bucle.

- Nuevo valor de consigna.

## 1.8. devices/control\_system/filters/pid

Este módulo es una sencilla aplicación de un filtro PID. Este módulo por supuesto solo puede ser utilizado como filtro de corrección.

Sigue esta ecuación:

$$out(i) = K_p e(i) + K_i T \sum_i e_i + K_d [e(i) - e(i - 1)]/T$$

$out(i)$  = Salida de control en iteración i

$K_p$  = Contante proporcional. Actua sobre el error de manera proporcional a este.

$K_i$  = Constante integral. Reduce el error acero en régimen permanente.

$K_d$  = Constante derivativa. Incrementa la estabilidad de la respuesta, reduciendo el sobreimpulso.

$e(i)$  = Entrada de error en iteración i

$T$  = Tiempo entre iteraciones i - 1

Esta ecuación discreta ha sido implementada con las siguientes particularidades:

- La constante integral (Ki) tiene que tener preintegrado el producto con el periodo de muestreo T. Cuando sea definida habrá que tenerlo en cuenta.
- Aunque el periodo del sistema de control de bucle es fijo, es posible definir un periodo aumentado para el cálculo derivativo mediante la definición de derivate\_nb\_samples.

Este sumatorio sigue la siguiente ecuación:

$$Kd [e(i) - e(i - n)]/n$$

$n = (\text{derivate}_{nb\_samples})$  numero de iteraciones del filtro, entre el que es calculado la diferencia de error.

Por defecto el valor de derivate\_nb\_samples es 1, con lo que la diferencia del error se calculará entre el error de la iteración actual y la inmediatamente anterior.

- Se han definido 3 variables configurables de nivel de saturación que limitarán el valor máximo que pueda tomar la entrada de error, el sumatorio de los errores acumulados para el cálculo del error integral y el valor de salida.

La ecuación programada es la siguiente:

```
command = in * p->gain_P + p->integral * p->gain_I + (derivate * p
> gain_D) / p->derivate_nb_samples
```

La estructura de datos contiene las siguientes variables:

- <gain\_P> Constante proporcional
- <gain\_I> Constante integral
- <gain\_D> Constante derivativa

- <out\_shift> Divisor en potencias de dos de la salida.

$$out(i) = out(i)/2^{out\_shift}$$

- <derivate\_nb\_samples> Número de iteraciones de filtro para cálculo de error derivativo.
- <index> Índice de buffer circular de errores acumulados. Variable para uso interno.
- <prev\_samples[PID\_DERIVATE\_FILTER\_MAX\_SIZE]> Array de buffer circular de errores acumulados. Variable para uso interno. El tamaño de este array está definido en el archivo de configuración “pid\_config.h” mediante la macro “PID\_DERIVATE\_FILTER\_MAX\_SIZE” cuyo valor por defecto es 4.
- <max\_in> Valor de saturación de señal de entrada.
- <max\_I> Valor de saturación de sumatorio de errores para error integral.
- <max\_out> Valor de saturación de señal de salida, una vez reducida con “out\_shift”.
- <Integral> Sumatorio de todos los errores de entrada desde que se inicializó el filtro. Variable para uso interno.
- <prev\_D> Valor derivado en iteración anterior sin Kd. Variable para uso interno.
- <prev\_out> Salida del proceso en la iteración anterior. Variable para uso interno.

Para simplificar el análisis y facilitar la comprensión tan solo se van a comentar aquellas funciones que han sido utilizadas en el código creado o son utilizadas en los módulos que han sido adaptados para el control de la electrónica dedicada a los sistemas de manipulación de elementos de juego.

- int32\_t **pid\_do\_filter**(void \*p, int32\_t in)  
Función de ejecución de filtro pid.

*Argumentos:*

- Puntero a estructura de filtro pid.
- Valor de entrada de error de filtro.

*Retorna:* Valor de salida de filtro.

- void **pid\_set\_gains**(struct pid\_filter \*p, int16\_t gp, int16\_t gi, int16\_t gd)  
Esta función ajusta las constantes del filtro PID.

*Argumentos:*

- Puntero a estructura de filtro pid.
- Nuevo valor de Kp.
- Nuevo valor de Ki.
- Nuevo valor de Kd.

- void **pid\_set\_maximums**(struct pid\_filter \*p, int32\_t max\_in, int32\_t max\_I, int32\_t max\_out)

Esta función ajusta las constantes de saturación del filtro PID.

*Argumentos:*

- Puntero a estructura de filtro pid.

- Nuevo valor de saturación de señal de entrada.
  - Nuevo valor de saturación de sumatorio de errores para error integral.
  - Nuevo valor de saturación de señal de salida, una vez reducida con “out\_shift”.
- void **pid\_set\_out\_shift**(struct pid\_filter \*p, uint8\_t out\_shift)  
*Esta función ajusta el divisor en potencias de dos de la salida.*  
*Argumentos:*
    - Nuevo valor de out\_shift.
  - int8\_t **pid\_set\_derivate\_filter**(struct pid\_filter \*p, uint8\_t nb\_samples)  
*Esta función ajusta el número de iteraciones de filtro para cálculo de error derivativo.*  
*Argumentos:*
    - Puntero a estructura de filtro pid.
    - Nuevo valor de derivate\_nb\_samples.*Retorna:* 0 si la asignación ha sido correcta y -1 si el número de iteraciones es mayor que el tamaño del buffer circular de almacenamiento de errores de entrada, de iteraciones anteriores.
- El acceso a las distintas variables de la estructura de datos del filtro PID, ha de hacerse a través de las siguientes funciones. Estas funciones deshabilitan las interrupciones antes de acceder a las variables para evitar la posible corrupción de datos. Todas estas funciones reciben como parámetro un puntero a la estructura del filtro y devuelven el valor de la variable solicitada:
- int16\_t **pid\_get\_gain\_P**(struct pid\_filter \*p)  
*Argumentos:*
    - Puntero a estructura de filtro PID.*Retorna:* Valor de Kp.
  - int16\_t **pid\_get\_gain\_I**(struct pid\_filter \*p)  
*Argumentos:*
    - Puntero a estructura de filtro PID.*Retorna:* Valor de Ki.
  - int16\_t **pid\_get\_gain\_D**(struct pid\_filter \*p)  
*Argumentos:*
    - Puntero a estructura de filtro PID.*Retorna:* Valor de Kd.
  - int32\_t **pid\_get\_max\_in**(struct pid\_filter \*p)  
*Argumentos:*
    - Puntero a estructura de filtro PID.*Retorna:* Valor de saturación de señal de entrada (max\_in).

- `int32_t pid_get_max_I(struct pid_filter *p)`

*Argumentos:*

- Puntero a estructura de filtro PID.

*Retorna:* Valor de saturación de sumatorio de errores para error integral (`max_I`).

- `int32_t pid_get_max_out(struct pid_filter *p)`

*Argumentos:*

- Puntero a estructura de filtro PID.

*Retorna:* Valor de saturación de señal de salida, una vez reducida con “`out_shift`” (`max_out`).

- `uint8_t pid_get_out_shift(struct pid_filter *p)`

*Argumentos:*

- Puntero a estructura de filtro PID.

*Retorna:* Valor de divisor en potencias de dos de la salida (`out_shift`).

- `uint8_t pid_get_derivate_filter(struct pid_filter *p)`

*Argumentos:*

- Puntero a estructura de filtro PID.

*Retorna:* Valor de número de iteraciones de filtro para calculo de error derivativo (`derivate_nb_samples`).

## 1.9. devices/control\_system/filters/quadramp

Este módulo implementa un filtro de consigna que limita la máxima variación a la salida del filtro entre dos iteraciones consecutivas, estableciendo un límite positivo y otro negativo de “consigna por periodo”, y también establece el incremento de la consigna de salida, definiendo un valor positivo y otro negativo de “incremento de consigna por periodo”. Ambas limitaciones corresponden a la derivada de primer y segundo orden de la consigna de entrada, la velocidad y a aceleración del sistema. Este módulo por supuesto solo puede ser utilizado como filtro de consigna.

El principio de funcionamiento se ha esquematizado en el Diagrama 1-2.

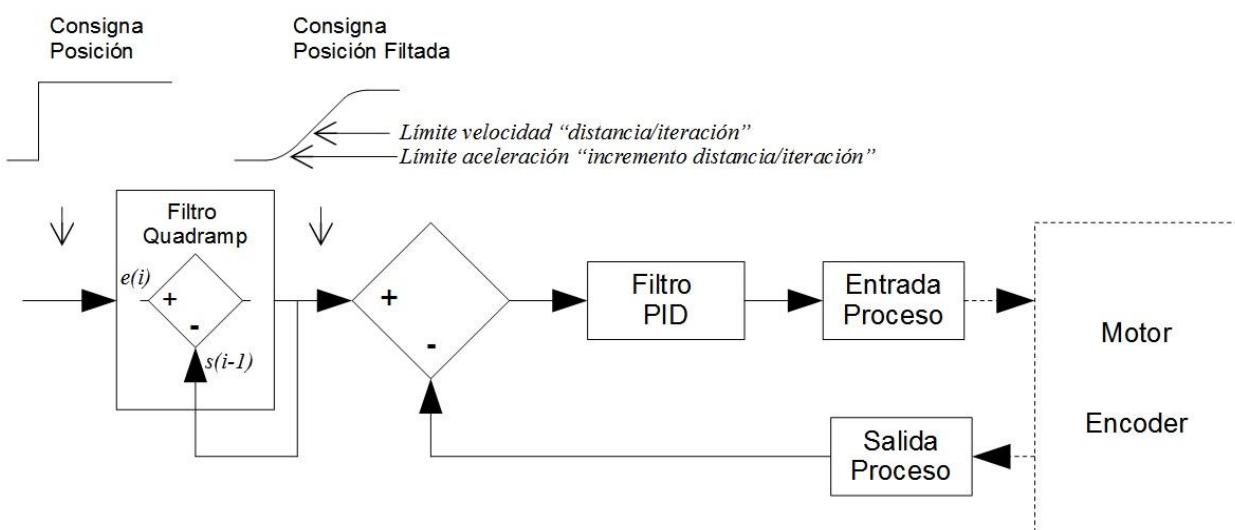


Diagrama 1-2 Diagrama funcionamiento filtro Quadramp

Como se puede ver este filtro se utilizará en sistemas de control de posición en los que tanto la consigna como las demás señales entre módulos corresponderán a valores de posición.

Otra particularidad de este filtro como puede verse en el esquema, es que no recibe realimentación real de la posición sino que para calcular la próxima consigna de posición utiliza la consigna de salida del filtro de la iteración anterior. Esto tiene las siguientes consecuencias:

- La velocidad y aceleración tendrán que ajustarse a valores inferiores de la velocidad y aceleración máximas de respuesta del sistema. En caso contrario se producirá la saturación del filtro PID.
- Como consecuencia de esto último la consigna de salida del filtro se convierte en la consigna destino de la iteración, siendo alcanzada por el sistema antes de la próxima iteración.

- Como consecuencia también de lo anterior los límites de velocidad, son definidos como distancia/iteración (habitualmente pulsos de encoder/iteración), y la aceleración aplicada como incremento de distancia/iteración (habitualmente incremento de pulsos de encoder/iteración).
- Si se produce un bloqueo mecánico, la consigna a la entrada del filtro PID cambiaria bruscamente, produciendo respuestas del sistema solo limitadas por las variables de saturación del filtro PID. Este sin duda es el mayor inconveniente de este filtro.

En el Diagrama 1-3 se muestra la respuesta del filtro.

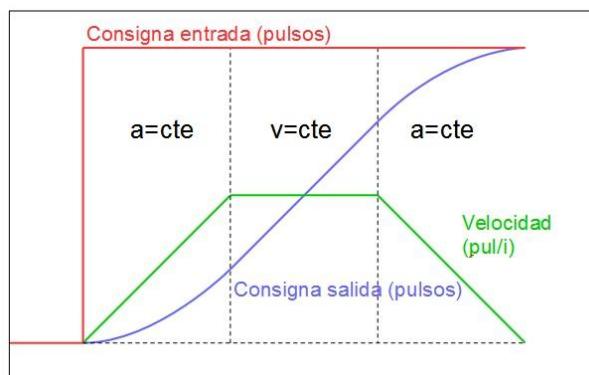


Diagrama 1-3 *Gráfica respuesta a consigna escalón de filtro Quadramp*

Como se ve en este esquema, este filtro no solo genera una rampa en el arranque del movimiento sino que internamente calcula el momento en el que tiene que reducir la velocidad aplicando la deceleración fijada, para manteniendo la rampa de parada generada alcanzar la posición de destino con velocidad nula.

La estructura de datos “quadramp\_filter” creada para este tipo de filtro contiene las siguientes variables:

- <var\_1nd\_ord\_pos> Límite de velocidad positiva en “distancia/iteración”. Limita la máxima variación positiva a la salida del filtro entre dos iteraciones consecutivas. Para el cálculo de esta variable será necesario conocer el periodo de ejecución del bucle de control.
- <var\_1nd\_ord\_neg> Límite de velocidad negativa en “distancia/iteración”. Limita la máxima variación negativa a la salida del filtro entre dos iteraciones consecutivas. Para el cálculo de esta variable será necesario conocer el periodo de ejecución del bucle de control.
- <var\_2st\_ord\_pos> Límite de aceleración positiva en “incremento de distancia/iteración”. Limita el incremento máximo positivo en la consigna de salida.
- <var\_2st\_ord\_neg> Límite de aceleración negativa en “decrementos de distancia/iteración”. Limita el incremento máximo negativo en la consigna de salida.

- <previous\_var> Velocidad en la iteración anterior. Variable para uso interno.
- <previous\_out> Consigna a la salida del filtro en la iteración anterior. Variable para uso interno.
- <previous\_in> Consigna a la entrada del filtro en la iteración anterior. Variable para uso interno.

Para simplificar el análisis y facilitar la comprensión tan solo se van a comentar aquellas funciones que han sido utilizadas en el código creado o son utilizadas en los módulos que han sido adaptados para el control de la electrónica dedicada a los sistemas de manipulación de elementos de juego.

- int32\_t **quadramp\_do\_filter**(void \* data, int32\_t in)  
Función de ejecución de filtro quadramp.

*Argumentos:*

- Puntero a estructura de filtro quadramp.
- Valor de entrada de consigna de filtro.

*Retorna:* Valor de salida de filtro.

- uint8\_t **quadramp\_is\_finished**(struct quadramp\_filter \*q)  
Esta función comprueba si el filtro ha concluido una trayectoria.

*Argumentos:*

- Puntero a estructura de filtro quadramp.

*Retorna:* 1 si se ha finalizado y 0 en caso contrario.

- void **quadramp\_init**(struct quadramp\_filter \* q)

Esta función resetea la estructura de datos apuntada por “quadramp\_filter”.

*Argumentos:*

- Puntero a estructura de filtro quadramp.

- void **quadramp\_reset**(struct quadramp\_filter \* q)

Esta función pone a cero las variables internas previous\_var, previous\_out y previous\_in.

*Argumentos:*

- Puntero a estructura de filtro quadramp.

- void **quadramp\_set\_2nd\_order\_vars**(struct quadramp\_filter \* q, uint32\_t var\_2nd\_ord\_pos, uint32\_t var\_2nd\_ord\_neg)

Esta función ajusta el valor de las variables de aceleración del filtro.

*Argumentos:*

- Puntero a estructura de filtro quadramp.
- Nuevo valor de aceleración positivo.
- Nuevo valor de aceleración negativo

- void **quadramp\_set\_1st\_order\_vars**(struct quadramp\_filter \* q, uint32\_t var\_1st\_ord\_pos, uint32\_t var\_1st\_ord\_neg)

Esta función ajusta el valor de las variables de límite de velocidad del filtro.

*Argumentos:*

- Puntero a estructura de filtro quadramp.
- Nuevo valor de límite de velocidad positivo.
- Nuevo valor de límite de velocidad negativo.

## **1.10. devices/robot/blocking\_detection\_manager**

Este módulo se utiliza para detectar el bloqueo del giro de motores, mediante la estimación de la corriente que están consumiendo, previa definición del umbral de corriente máximo y velocidad mínima.

El motor se encontrara bloqueado cuando se cumplan estas dos condiciones:

- La corriente que está consumiendo se encuentra por encima del umbral definido durante más de un número determinado de iteraciones.
- La velocidad de giro del motor está por debajo del umbral definido. En caso de que no sea definido no se tendrá en cuenta esta.

El consumo de corriente del motor no es medido sino que es estimado mediante la siguiente fórmula:

$$i = k1V - k2w$$

Donde V es el voltaje aplicado al motor y W es la velocidad medida de giro en este.

Acudiendo al circuito equivalente de un motor de corriente continua y a su ecuación, se deduce con facilidad la fórmula aplicada, teniendo en cuenta que la tensión contraelectromotriz inducida en el devanado es directamente proporcional a la velocidad de giro del motor.

$$V_i = V_e + Ri(t) + Ldi(t)/dt$$

$$\text{Simplificando } di(t)/dt = 0 \text{ (Regimen permanente)}$$

$$i(t) = V_i/R - V_e/R$$

$$V_e = K_b w(t)$$

$$i(t) = (1/R)V_i - (K_b/R)w(t)$$

$$K1 = (1/R)$$

$$K2 = (K_b/R)$$

$$i(t) = K1V_i - K2w(t)$$

*Siendo:*

$V_i$  = Tensión de alimentación del motor.

$V_e$  = Tensión contraelectromotriz generada en el estator.

$R$  = Resistencia del devanado.

$L$  = Inductancia del devanado.

$K_b$  = Contante contraelectromotriz.

$w(t)$  = Velocidad de giro del motor.

La estructura de datos “blocking\_detection” creada para este módulo contiene las siguientes variables:

- <cs> Puntero a la estructura de control de bucle sobre la actúa el control de bloqueo del motor.
- <k1> Constante  $K_1 \approx 1/R$
- <k2> Constante  $k_2 \approx K_b/R$
- <cpt\_thres> Número de iteraciones que tiene que estar la corriente por encima del umbral de corriente para considerarse motor bloqueado.
- <cpt> Variable que almacena el número de iteraciones que la corriente superó el umbral de corriente. Variable de uso interno.
- <speed\_thres> Umbral de velocidad del motor por encima del cual no se detecta el bloqueo del motor. Si no es definido no hay límite de velocidad mínimo, el detector de bloqueo funciona a cualquier velocidad (speed\_thres=0).
- <prev\_pos> Posición previa (encóder) en anterior iteración. Es utilizada para calcular la velocidad del motor.

Para simplificar el análisis y facilitar la comprensión tan solo se van a comentar aquellas funciones que han sido utilizadas en el código creado o son utilizadas en los módulos que han sido adaptados para el control de la electrónica dedicada a los sistemas de manipulación de elementos de juego.

- **void bd\_manage\_from\_cs(struct blocking\_detection \* bd, struct cs \*cs)**  
Función de ejecución de sistema del control de bloqueo sobre el sistema de control de bucle definido por la estructura apuntada por “cs”. Esta función tiene que ser llamada periódicamente.  
Esta función internamente accede a la estructura de datos del control de bucle (mediante las funciones de acceso) para leer la salida del filtro de corrección, que corresponde a la tensión que se aplica al motor, y la salida del filtro de realimentación, que corresponde a la posición del sistema.  
Con esta posición y la previa que guarda en la variable “prev\_pos”, calcula la velocidad de giro del motor.  
Con la velocidad calculada y la consigna de tensión que se aplica al motor, calcula según la fórmula que se ha explicado la velocidad de giro. Y ya con esta velocidad, y los límites establecidos determina si se ha producido un bloqueo del motor. En caso de bloqueo se incrementa un contador de períodos de ejecución en bloqueo bd->cpt hasta el umbral fijado para esta variable.

Las acciones a realizar en caso de bloqueo tienen que practicarse desde funciones externas a este módulo que comprueben el estado de la variable “bd->cpt” y actúen según el caso.

*Argumentos:*

- Puntero a estructura de detección de bloqueo.
- Puntero a estructura del control de bucle sobre el que se aplica el control de bloqueo.

- void **bd\_set\_current\_thresholds**(struct blocking\_detection \* bd, int32\_t k1, int32\_t k2, uint32\_t i\_thres, uint16\_t cpt\_thres)

Esta función ajusta todas las variables del bloqueo de control de bloqueo, salvo “speed\_thres”.

*Argumentos:*

- Puntero a estructura de detección de bloqueo.
- Constante K1  $\approx 1/R$ .
- Constante k2  $\approx K_b/R$ .
- Umbral de corriente.
- Iteraciones de umbral de corriente.

- void **bd\_init**(struct blocking\_detection \* bd)

Esta función resetea la estructura de datos apuntada por “blocking\_detection”. Pone todos los bits a cero.

*Argumentos:*

- Puntero a estructura de detección de bloqueo.

- void **bd\_set\_speed\_threshold**(struct blocking\_detection \* bd, uint16\_t speed)

Esta función ajusta un nuevo valor para el umbral de velocidad.

*Argumentos:*

- Puntero a estructura de detección de bloqueo.
- Nueva velocidad umbral.

- void **bd\_reset**(struct blocking\_detection \* bd)

Esta función pone a cero el contador de iteraciones consecutivas en corriente superada.

*Argumentos:*

- Puntero a estructura de detección de bloqueo.

- uint8\_t **bd\_get**(struct blocking\_detection \* bd)

Esta función devuelve “1” si el umbral de límite de iteraciones consecutivas en corriente superada ha sido configurado y ha sido alcanzado dicho umbral. “0” en caso contrario.

*Argumentos:*

- Puntero a estructura de detección de bloqueo.

*Retorna:* 1 si el motor está bloqueado y cero si no lo está.

## **1.11. devices/encoders\_dspic**

---

Este módulo permite configurar y controlar hasta dos entradas de encoder. Y permite leer la posición absoluta de cada uno de los encoder habilitados. Tiene una función de control llamada “encoder\_dspic\_manage” que tiene que ser ejecutada periódicamente, siendo lo habitual ejecutarla junto con las funciones de control de bucle y la de detección de bloqueo. Esto es a la misma frecuencia.

Este módulo no tiene una estructura de datos asociada, tan solo dispone de un array “g\_encoders\_dspic\_values[ENCODERS\_DSPIC\_NUMBER]” donde se almacena siempre la última posición absoluta calculada de cada encoder. Existe otro array donde se almacena la última lectura del registro del encoder, y se utiliza para poder calcular el valor absoluto sin desbordamiento en la siguiente iteración.

Al igual que en los otros módulos estas variables no deben ser accedidas directamente sino a través de las funciones creadas para ese cometido. En estas funciones se accede a estas variables en zonas de código libres de interrupciones para evitar la corrupción de datos.

Para simplificar el análisis y facilitar la comprensión tan solo se van a comentar aquellas funciones que han sido utilizadas en el código creado o son utilizadas en los módulos que han sido adaptados para el control de la electrónica dedicada a los sistemas de manipulación de elementos de juego.

- **void encoders\_dspic\_manage(void \* dummy)**

Función de ejecución de sistema de encóder, en esta función se actualiza la posición absoluta del encóder. Tiene que ser ejecutada periódicamente.

*Argumentos:*

- Puntero sin utilidad. NULL.

- **void encoders\_dspic\_init(void)**

Esta función configura y arranca el encóder 1 y 2. Previamente es necesario definir:

- ENCODERS\_DSPIC1\_ENABLED para habilitar el encóder 1.
  - ENCODERS\_DSPIC2\_ENABLED para habilitar el encóder 2.

También resetea a cero la posición absoluta del encóder

- **int32\_tencoders\_dspic\_get\_value(void \* number)**

Esta función devuelve el valor absoluto del número de encóder que recibe como parámetro.

*Argumentos:*

- Número de encóder para leer con este formato “(void \*)1”, internamente se transforma a una variable de tipo int con una conversión cast.

*Retorna:* El valor absoluto de la última lectura del encóder

- **voidencoders\_dspic\_set\_value(void \* number, int32\_t v)**

Esta función ajusta el valor de la variable absoluta del encóder

*Argumentos:*

- Número de encóder para leer con este formato “(void \*)1”, internamente se transforma a una variable de tipo int con una conversión cast.
- Nuevo valor para la variable de posición absoluta del encóder

## 1.12. devices/servo/ax12

Este módulo permite controlar los servos modelo AX12 de la marca Dynamixel. Los servos AX12 son unos servos digitales diseñados específicamente para aplicaciones de robótica.

### 1.12.1. Protocolo de comunicación y funciones de acceso a registros

Su protocolo de comunicación es serie asíncrono half duplex con 8 bits de datos, 1 bit de stop y sin paridad. Siendo la conexión física de niveles TTL Multi Drop.

La conexión TTL Multi Drop permite la conexión de muchos dispositivos, a una misma línea de comunicación. Esta línea es conectada a 5V a través de una resistencia de Pull-Up, y en los dispositivos a una salida configurada en colector abierto. De esta forma el dispositivo que tome el control de la línea enviará los mensajes conectando la línea a masa. Si varios dispositivos intentan transmitir al mismo tiempo, los mensajes se machacarán unos a otros. Para impedir esto existe un mecanismo de arbitraje que en el caso de los servos AX12 es muy simple, al comportarse todos los dispositivos como esclavos de la tarjeta de control.

Por otro lado para la conexión de la linea half duplex (comunicación bidireccional por una línea) de los Dynamixel a la uart se recomienda la utilización del circuito de control CM-5 de Dynamixel. Este circuito tiene el esquema interno de funcionamiento mostrado en la Figura 1-1.

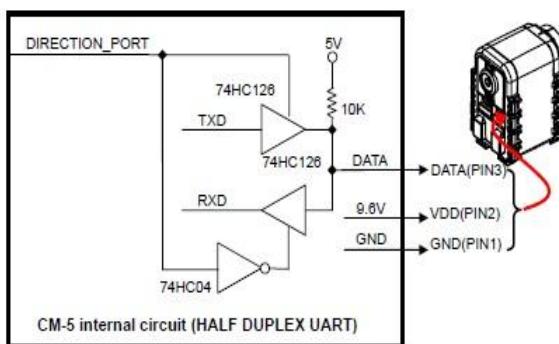


Figura 1-1 Conexión a UART a través de circuito CM-5 de Dynamixel (User's Manual Dynamixel AX-12, 2006)

De este módulo salen tres líneas a la tarjeta de control. Txd y Rxd que se conectan a la uart y Direction\_Port que con un nivel alto, conecta Txd a la linea de comunicación half duplex DATA, y con un nivel bajo Rxd a dicha línea.

Como se va a explicar un poco más adelante, las librerías creadas permiten definir una función para comunicar la parte de estas de más bajo nivel con el hardware. Y esto nos va a permitir la utilización con facilidad de otra estructura de conexión de salida diferente.

El protocolo de comunicación entre la tarjeta de control y los AX12 conectados a la línea de control sigue siempre la misma estructura. La tarjeta comienza la comunicación enviando un paquete de datos de instrucción, esta comunicación es procesada por todos los AX12 conectados a la línea, pero solo es atendida por el servo con el ID incluido en el paquete de datos. Este responde con otro paquete de datos con una estructura diferente, y denominado de “status”. Existe una excepción, que es cuando la instrucción es enviada a todos los AX-12 conectados, mediante el ID “0xFE”, en cuyo caso no hay respuesta.

La estructura del paquete de instrucción es el siguiente:

0xFF+0xFF+ID+LENGTH+INSTRUCTION+PARAMETER1...PARAMETER\_N+CHECK\_SUM

Siendo:

- 0xFF+0xFF Inicio de paquete
- ID Identificador del AX12 al que va dirigido el paquete.
- LENGTH Longitud del paquete calculada como “Número de parámetros (N) + 2”.
- INSTRUCTION número de instrucción enviada.
- PARAMETER0..N Utilizados según la instrucción enviada.
- CHECK\_SUM Valor de suma de verificación de paquete enviado. Se calcula siguiendo esta fórmula: Check Sum = ~ (ID + Length + Instruction + Parameter1 + ... Parameter N). En caso de desbordamiento del byte, este no será tenido en cuenta. Finalmente se aplica una operación NOT a nivel de bits. Este valor resultante es utilizado por el AX-12 para comprobar posibles cambios accidentales en la secuencia de datos recibida. Calculando nuevamente la suma de verificación y comprobando la igualdad con el valor recibido.

La estructura del paquete de status es el siguiente:

0xFF+0xFF+ID+LENGTH+ERROR+PARAMETER1...PARAMETER\_N+CHECK\_SUM

Siendo:

- 0xFF+0xFF Inicio de paquete
- ID Identificador de la tarjeta que envío el paquete de instrucción. Por defecto este está configurado en los AX-12 como número “1”.
- LENGTH Longitud del paquete calculada como “Número de parámetros (N) + 2”.

- ERROR Error ocurrido en AX12. En este byte se codifican 7 errores, uno por cada bit, desde el de menor peso.
- PARAMETER0..N Utilizados si son necesarios.
- CHECK\_SUM Valor de suma de verificación de paquete enviado. Se calcula siguiendo esta fórmula: Check Sum = ~ (ID + Length + Instruction + Parameter1 + ... Parameter N). En caso de desbordamiento del byte, este no será tenido en cuenta. Finalmente se aplica una operación NOT a nivel de bits. Este valor resultante es utilizado por el AX-12 para comprobar posibles cambios accidentales en la secuencia de datos recibida. Calculando nuevamente la suma de verificación y comprobando la igualdad con el valor recibido.

La lista de instrucciones se muestra en la Tabla 1-1, junto con su correspondencia con la macro definida en las Aversive:

Instrucción	Función	Valor	Macro Aversive
PING	Ninguna acción. Utilizado para recibir el paquete de status.	1	AX12_PING
READ DATA	Lectura de valores de la tabla de control.	2	AX12_READ
WRITE DATA	Escrutura de valores de la tabla de control.	3	AX12_WRITE
REG WRITE	Misma acción que WRITE DATA pero se mantiene en espera hasta una nueva instrucción de "ACTION"	4	AX12_REG_WRITE
ACTION	Disparador para la acción registrada por "REG WRITE"	5	AX12_ACTION
RESET	Resetea todos los registros de la memoria eeprom al valor de fábrica.	6	AX12_RESET

Tabla 1-1 *Juego de instrucciones de control de servos Dynamixel AX12* (Robotis, 2006)

De todas estas instrucciones solo se va a emplear las de lectura y escritura sencillas.

La estructura del paquete de datos de una instrucción de escritura en un registro será la siguiente:

- Longitud (N+3) + Instrucción (0x03) + Parametro1 (Comienzo de la dirección de escritura) + Parametro2 (Primer byte de datos)+ ... + ParametroN+1 (N byte de datos)

Y la estructura del paquete de datos de una instrucción de escritura en un registro será la siguiente:

- Longitud (0x04) + Instrucción (0x02) + Parametro1 (Comienzo de la dirección de lectura) + Parametro2 (Bytes del dato a ser leído)

En ambos casos se han obviado los bytes de comienzo de paquete y el de suma de verificación.

Todos los registros accesibles en los AX-12 se encuentran codificados en el archivo “ax12\_address.h” mediante la definición de una variable de tipo enumerado y la declaración de la variable “AX12\_ADDRESS”.

Este módulo ha sido estructurado por niveles muy fácilmente identificables según su nivel de abstracción. Se han creado toda una serie de funciones que van accediendo secuencialmente a más bajo nivel hasta alcanzar el hardware.

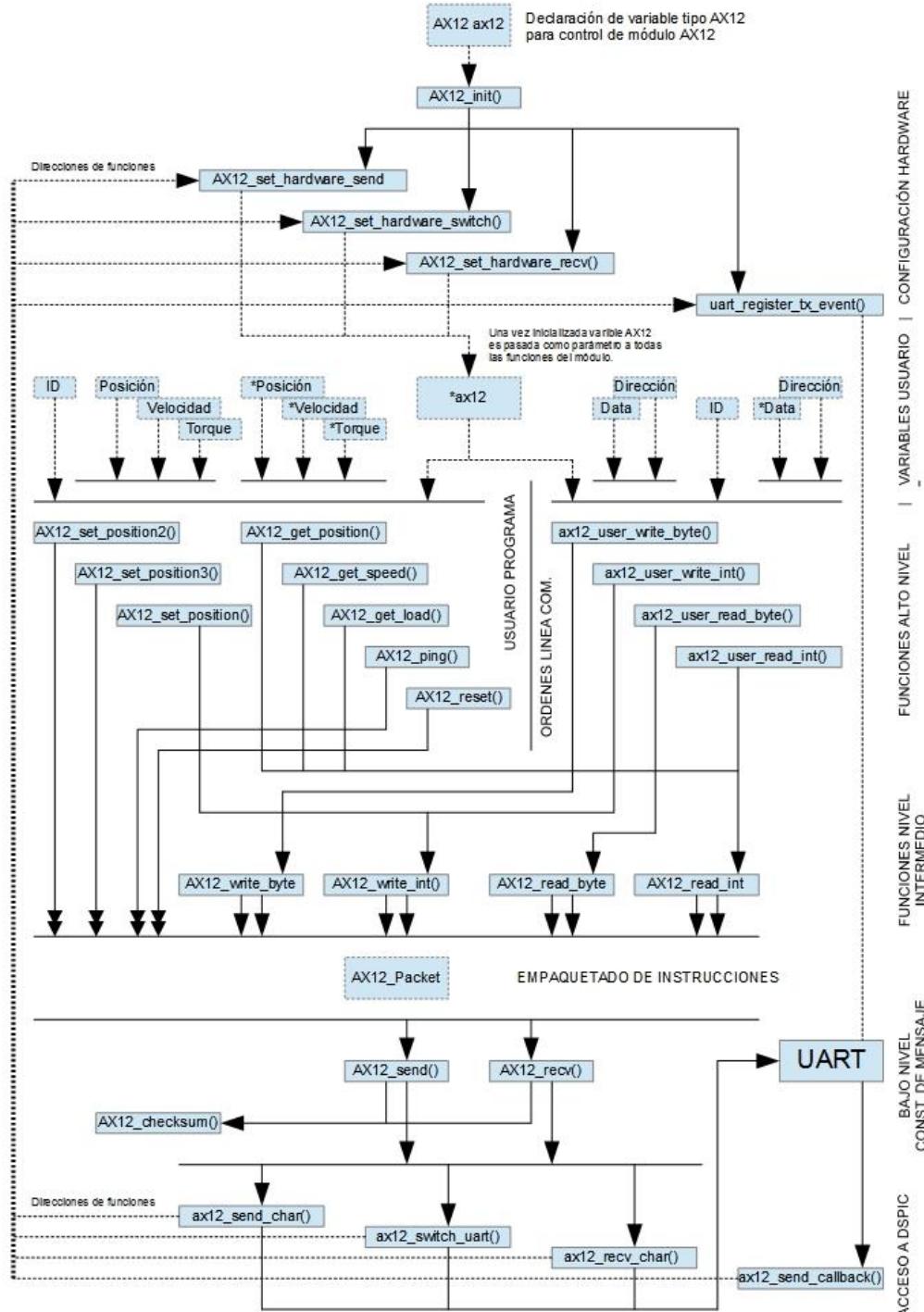


Diagrama 1-4 Diagrama de flujo de funciones de módulo AX12

### 1.12.2. Inicialización y configuración de la interfaz HW

---

Antes de utilizar el módulo es necesario indicar a este que funciones van a hacer de enlace entre el protocolo full duplex y el half duplex. Para almacenar esta información existe una variable AX12 de tipo estructura que contiene tres punteros a funciones, donde se almacenan las direcciones de las funciones que se ejecutarán para el envío de un carácter a través del hardware half duplex, la recepción y el cambio de la dirección del bus respectivamente. La dirección de esta variable será pasada recurrentemente de una función a otra del módulo para la utilización de estas funciones en el más bajo nivel.

También es necesario registrar un evento de ejecución de una función cuando hayan sido enviados todos los bits en el registro de salida del puerto serie que vaya a ser utilizado para la comunicación. Esto es necesario debido a que se puede dar la circunstancia de que se cambie de modo de emisión a recepción, sin que se haya terminado el envío de todos los bits. Y como se explicará más adelante en esta función de evento de final de envío de byte se cambiara automáticamente a modo de recepción, ya que después de una emisión siempre habrá una recepción.

Describo una a una estas funciones ya que serán utilizadas en la programación del robot:

- **void AX12\_init(AX12 \*s)**  
Esta función borra todo el contenido a la estructura apuntada por ax12.  
*Argumentos:*
  - Puntero una estructura AX12.
  
- **void AX12\_set.hardware\_send(AX12 \*ax12, int8\_t(\*)(uint8\_t))**  
Esta función almacena en un puntero a la estructura apuntada por ax12 la dirección de la función que será invocada cuando se quiera enviar un carácter en protocolo half duplex.  
*Argumentos:*
  - Puntero una estructura AX12.
  - Dirección de función.
  
- **void AX12\_set.hardware\_send(AX12 \*ax12, int8\_t(\*)(uint8\_t))**  
Esta función almacena en un puntero a la estructura apuntada por ax12 la dirección de la función que será invocada cuando se quiera enviar un carácter en protocolo half duplex.  
*Argumentos:*
  - Puntero una estructura AX12.
  - Dirección de función.

- **void AX12\_set\_hardware\_recv(AX12 \*ax12, int16\_t(\*)(void))**

Esta función almacena en un puntero a la estructura apuntada por ax12 la dirección de la función que será invocada cuando se quiera recibir un carácter en protocolo half dúplex

*Argumentos:*

- Puntero una estructura AX12.
- Dirección de función.

- **void AX12\_set\_hardware\_switch(AX12 \*ax12, void(\*)(uint8\_t))**

Esta función almacena en un puntero a la estructura apuntada por ax12 la dirección de la función que será invocada cuando se quiera cambiar la línea de comunicación utilizada en protocolo half dúplex de modo de transmisión. Existirán dos modos posibles, emisión y recepción.

*Argumentos:*

- Puntero una estructura AX12.
- Dirección de función.

Aunque esta función pertenece al módulo UART, no fue explicada en el apartado correspondiente a dicho módulo.

- **void uart\_register\_tx\_event(uint8\_t num, void (\*f)(char))**

Esta función programa la ejecución de una función cuando sea concluida la transmisión de todos los bits del registro asociado al número de uart.

*Argumentos:*

- Número de la uart sobre la que se registrara el evento de transmisión.
- Dirección de la función que será ejecutada cuando sea concluida la transmisión de todos los bits del registro asociado a la uart.

---

### 1.12.3. Funciones de alto nivel

---

Estas funciones son las de usuario de la librería, y constituyen el mayor nivel de abstracción sobre el control de los servos AX12. Pero tan solo han sido creadas las funciones para las instrucciones más habituales. Estas funciones reciben además de la estructura del módulo, el ID del servo y los parámetros de las magnitudes físicas que se quieren modificar o leer. Posición, velocidad y torque.

Las funciones de este nivel pueden acceder tanto a funciones de nivel intermedio como a funciones de bajo nivel como se puede ver en el diagrama de flujo. Esto se explica por qué la funciones de nivel intermedio solo están preparadas para enviar y leer 1 o 2 byte, según sean char o int lo datos. Y hay funciones de alto nivel que necesitan acceder a más posiciones de memoria consecutivas. O también a casos en los que no se accede a ninguna posición. En estos casos se accede directamente a través de las funciones de bajo nivel.

Describo una a una solo las funciones que serán utilizadas en la programación del robot:

- **uint8\_tAX12\_set\_position(AX12 \*s,uint8\_t id, uint16\_t position)**  
Esta función ajusta el registro interno de posición del servo, esto provocara el desplazamiento del servo a la velocidad y torque configurado internamente en el servo.

*Argumentos:*

- Puntero a una estructura AX12.
- ID del servo a controlar
- Nueva posición del servo

*Retorna:* 0 si recepción correcta y ningún error en AX12, o código de error devuelto por AX12 en caso contrario.

- **uint8\_tAX12\_get\_position(AX12 \*ax12, uint8\_t id, uint16\_t \*pos)**  
Esta función lee el registro interno de posición del servo, que corresponde a la posición actual del servo.

*Argumentos:*

- Puntero a una estructura AX12.
- ID del servo a controlar
- Dirección de variable donde se almacenara el valor leído.

*Retorna:* 0 si lectura correcta y ningún error en AX12, o código de error devuelto por AX12 en caso contrario. En caso de recepción correcta el valor leído es cargado en la dirección apuntada por “pos”, en caso contrario mantiene inalterada dicha posición de memoria.

- **uint8\_t AX12\_ping(AX12 \*ax12, uint8\_t id)**  
Esta función no realiza ninguna acción en el interior del servo ID, tan solo pide un mensaje de respuesta del dicho servo.

*Argumentos:*

- Puntero a una estructura AX12.
- ID del servo al que va dirigido el PING.

*Retorna:* El registro de error devuelto en la respuesta del AX12.

## 1.13. hardware/dspic/pwm\_servo

Este módulo esta específicamente programado para el control de hasta 4 servos analógicos desde un dsPIC. El control se realiza desde cuatro salidas de comparador y permite seleccionar como base de tiempos el timer 2 o el 3. Para su selección existe un archivo llamado “pwm\_servo\_config.h” donde también es posible seleccionar la frecuencia de la señal PWM generada. Su valor por defecto es el siguiente:

- PWM\_FRE 50 (repeticiones/seg)
- TIMER\_SOURCE 2

En esta librería se ha definido un tipo de estructura “pwm\_servo” que contiene todos los datos para el control de cada uno de los cuatro servos. Contiene los siguientes datos:

- num número de la salida de comparación (Puede tomar valores desde el 1 hasta el 4)
- th\_min Valor mínimo de cuentas que puede ser asignado al registro OCnRS para comparación.
- th\_max Valor máximo de cuentas que puede ser asignado al registro OCnRS para comparación.
- range Este valor corresponde a “ th\_max - th\_min” (No tiene ninguna utilidad en las librerías)

Para la configuración de cada una de las salidas de comparación existe una función llamada “pwm\_servo\_init”. Esta función recibe como parámetros un puntero a la estructura “pwm\_servo” donde se almacenara la información de la salida configurada, el número de la salida a configurar, el tiempo en microsegundos a nivel alto de la señal PWM generada mínimo y el máximo.

Esta función abstrae completamente al programador del ajuste de los registros internos del módulo de comparación y del temporizador elegido mediante “TIMER\_SOURCE”, ya que también realiza la configuración de este

Internamente realiza las siguientes configuraciones y acciones:

- Rellena la variable “pwm\_servo” pasada por referencia para el control de esta salida.  
Para ello cambia de unidades th\_max y th\_min, de microsegundos a cuentas de registro de comparación.

$$th_{max} (\text{cuentas}) = th_{max} (\mu\text{s}) \times (1/1.000.000)(\text{s}/\mu\text{s}) \times (\text{TIMER_FREQ})(\text{cuentas/seg})$$

$$th_{min} (\text{cuentas}) = th_{min} (\mu\text{s}) \times (1/1.000.000)(\text{s}/\mu\text{s}) \times (\text{TIMER_FREQ})(\text{cuentas/seg})$$

- Configuración de salida de módulo de comparación. Se fija el modo de funcionamiento de pulso continuo. En este modo la salida se mantiene a nivel alto mientras el valor del Timer asociado se encuentre entre los valores almacenados en los registros de configuración OCxR y OcxRS.

Por defecto el valor del registro OCxR siempre es ajustado a 0 en esta función, por lo que será el valor del registro OCxRS el que se empleara para fijar el tiempo que la PWM permanece a nivel alto. Inicialmente este valor es ajustado en la mitad del rango entre th\_max y th\_min.

- Configuración del Timer asociado a la salida de comparación. Es necesario configurar el periodo de desbordamiento de este Timer que definirá el periodo de la señal PWM. Para esto es necesario ajustar el registro del periodo del timer “PRx” con el siguiente calculo:

- Calculo para dsPIC en tarjeta con cristal de 10Mhz.
- FOSC = 10MHz \* 4 = 40 MHz

- FCY = FOSC / 2 = 20MHz
- TIMER\_FREQ = FCY / TIMER\_DIV = 20MHz / 64 = 312500 cuentas/seg
- PRx = TIMER\_FREQ / PWM\_FREQ = 312500 (cuentas/seg) / 50 (periodosPWM/seg) = 6250 (cuentas / periodoPWM)

Este módulo dispone de las siguientes funciones:

- void **pwm\_servo\_init**(struct pwm\_servo \*pwm, uint16\_t num, uint16\_t th\_min\_us, uint16\_t th\_max\_us)

Esta función inicializa y habilita la salida número “num” del módulo de comparación de un dsPIC para el control de un servo analógico. Rellena de variable de tipo estructura “pwm\_servo” que es pasada por referencia como parámetro, con todos los datos de caracterización de la salida. Calcula el valor de las unidades th\_max y th\_min, de microsegundos a cuentas de registro de comparación. Y por último inicializa y habilita el timer que se utilizará como base de tiempos.

*Argumentos:*

- Puntero a una estructura pwm\_servo, dirección a la estructura de caracterización del servo.
- Número de la salida a configurar.
- Tiempo en microsegundos mínimo de la señal PWM a nivel alto.
- Tiempo en microsegundos máximo de la señal PWM a nivel alto.

- uint16\_t **pwm\_servo\_set**(struct pwm\_servo \*pwm, uint16\_t value)

Función para el ajuste de la posición de un servo analógico, tiene que recibir la dirección de la estructura de caracterización de la salida a utilizar y el nuevo valor de posición. Este valor será introducido en cuentas de temporizador, teniendo que estar comprendido entre 0 y th\_max.

*Argumentos:*

- Puntero a una estructura pwm\_servo, dirección a la estructura de caracterización del servo. Incluye el número de salida del servo.
- Nuevo valor de posición. Este valor será introducido en cuentas de temporizador, teniendo que estar comprendido entre 0 y th\_max.
- Dirección de variable donde se almacenará el valor leído

*Retorna:* El valor ajustado, ya que puede haber sido saturado al th\_max.

- void **pwm\_servo\_enable**(void)

Esta función habilita todas las salidas de servos, mediante la habilitación del temporizador asociado.

- void **pwm\_servo\_disable**(void)

Esta función deshabilita todas las salidas de servos, mediante la deshabilitación del temporizador asociado.

## 1.14. hardware/dspic/pwm\_mc

Esta librería simplifica las tareas de configuración y control, de los módulos “MCPWM” de salidas PWM múltiples sincronizadas de los dsPIC, para el manejo de controladoras de motores de diferentes tipos. Puede controlar hasta dos módulos, que es el número habitual implementado en los dPIC. Estos módulos tienen características y capacidades diferentes.

En esta librería se ha definido un tipo de estructura “pwm\_mc” que contiene todos los datos para el control de un motor a través del módulo MCPWM. Contiene los siguientes datos:

- pwm\_mc es una estructura donde se almacena la información del canal.
- pwm\_mode define el modo de funcionamiento. (PWM\_MC\_MODE\_XXX).
- pwm\_module\_num número de módulo DCMCPWM, puede ser el 1 o el 2.
- pwm\_channel\_num es el canal del módulo (El módulo 1 dispone de 3 canales, y el 2 de 1).
- sign\_port es el puerto donde está mapeado el pin “sing”.
- sign\_bit es el número del bit del pin sing en el puerto sing.
- sign\_port\_n es el puerto donde está mapeado el pin “sing” negativo.
- sign\_bit\_n es el número del bit del pin sing negativo en el puerto sing negativo.

Este módulo dispone de las siguientes funciones:

- **void pwm\_mc\_channel\_init(struct pwm\_mc \*pwm, uint8\_t pwm\_mode, uint8\_t pwm\_module\_num, uint8\_t pwm\_channel\_num, volatile uint16\_t \*sign\_port, uint8\_t sign\_bit, volatile uint16\_t \*sign\_port\_n, uint8\_t sign\_bit\_n)**  
Función para la inicialización de la estructura de configuración de un canal. Se almacenan en esta estructura todos los datos pasados como parámetros.

*Argumentos:*

- Puntero a la estructura donde será almacenada la configuración del canal a inicializar.
- Modo de funcionamiento. Los principales modos son Normal, Signed y Bipolar.
- Número del módulo. Puede ser el 1 (que corresponde al MCPWM=0) o el 2 (que corresponde al MCPWM=1), internamente en esta función se resta 1 al valor introducido.
- Canal del módulo (El módulo 1 dispone de 3 canales, y el 2 de 1).
- Es el puerto donde está mapeado el pin “sing”.
- Es el número del bit del pin sing en el puerto sing.
- Es el puerto donde está mapeado el pin “sing” negativo.
- Es el número del bit del pin sing negativo en el puerto sing negativo.

- **void pwm\_mc\_init(struct pwm\_mc \*pwm, uint16\_t f\_pwm, uint16\_t pwm\_pins\_config)**

Esta función inicializa la generación de las PWMs en un módulo a una frecuencia determinada y configura los pines de salida de canal.

Con esta configuración queda fijada por un lado la resolución en número de bits de la PWM y el valor máximo ajustable de pwm.

$$\text{Resolución}_{\text{pwm}} = \log_2 \left( 2 \times \frac{\text{FCY}}{f_{\text{PWM}}} \right); \text{ Siendo FCY} = \text{frecuencia cuarzo} \times 2$$

Dependiendo del modo de funcionamiento seleccionado para un canal los valores máximos son ajustados entre diferentes márgenes:

Siendo periodo = ((FCY / (f\_pwm \* prescaler))-1)

- Modo “PWM\_MC\_MODE\_SIGNED”:  
 $\text{pwm\_max[]} = 2 * \text{periodo}$   
 $\text{pwm\_min[]} = -2 * \text{periodo}$   
 $\text{pwm\_offset[]} = 0$
- Modo “PWM\_MC\_MODE\_BIPOLAR”  
 $\text{pwm\_max[]} = \text{periodo}-1$   
 $\text{pwm\_min[]} = -\text{periodo}$   
 $\text{pwm\_offset[]} = \text{periodo}$
- Modo “PWM\_MC\_MODE\_SIGNED” y resto de modos.  
 $\text{pwm\_max[]} = 2 * \text{periodo}$   
 $\text{pwm\_min[]} = 0$   
 $\text{pwm\_offset[]} = 0$

#### Argumentos:

- Puntero a una estructura pwm, donde se almacena la información del módulo a inicializar.
- La frecuencia que se fijara como frecuencia de PWM. Se introduce en Hz.
- Configuración de los pines de salida de los canales. Se codifican con una operación AND a nivel de bits con las macros codificadas en el archivo “pwm\_mc.h”. Un ejemplo sería:  
 Generación de PWM por canal 1 en modo complementario  
 (implica que canal L y canal H estén habilitados para salida de canal).  
 $\text{pwm\_pins\_config} = \text{MOD1\_COMP} \& \text{PEN1H} \& \text{PEN1L};$
- **void pwm\_mc\_set(void \*pwm, int32\_t value)**  
 Esta función ajusta el valor de salida de la PWM del canal definido en pwm.  
 Puntero a la estructura donde esta almacenada la configuración del canal.  
 Nuevo valor de pwm. Tendrá que estar entre los valores almacenados en  $\text{pwm\_max}[]$  y  $\text{pwm\_min}[]$ .



## **2. Conexionado hardware detallado**

---

En el capítulo de electrónica de control ya fueron descritas las conexiones de los sensores y actuadores a la tarjeta de control integrada. En este apartado se van a especificar todas las conexiones en los diferentes puertos del dsPIC33FJ128MC804, incluyendo ahora todo el conexionado interno con la parte de la tarjeta del microcontrolador maestro o de alto nivel. Estas conexiones se van a convertir en especificaciones de diseño ya que han venido impuestas por el diseño de la tarjeta.

### **2.1. Conexionado de puertos en dsPIC**

---

En la Figura 2-1 se ha esquematizado todo el conexionado de los pines del microcontrolador esclavo.

Los diferentes sensores y actuadores han sido agrupados por sistemas y subsistemas de manipulación. También se ha indicado si la conexión es directa a las salidas de la tarjeta, directa con algún tipo de adaptación (resistencias pull-up) o a través de algún driver integrado en la tarjeta.

Existen dos líneas de comunicación entre el micro esclavo y el maestro. La línea de comandos de maestro a esclavo a través de un puerto I2C (SCA y SCL) y la línea de comandos uart (Tx y Rx) de PC a esclavo. Esta línea pasa a través del micro maestro que redirige en ambos sentidos todo lo que recibe desde el PC hasta el esclavo y viceversa.

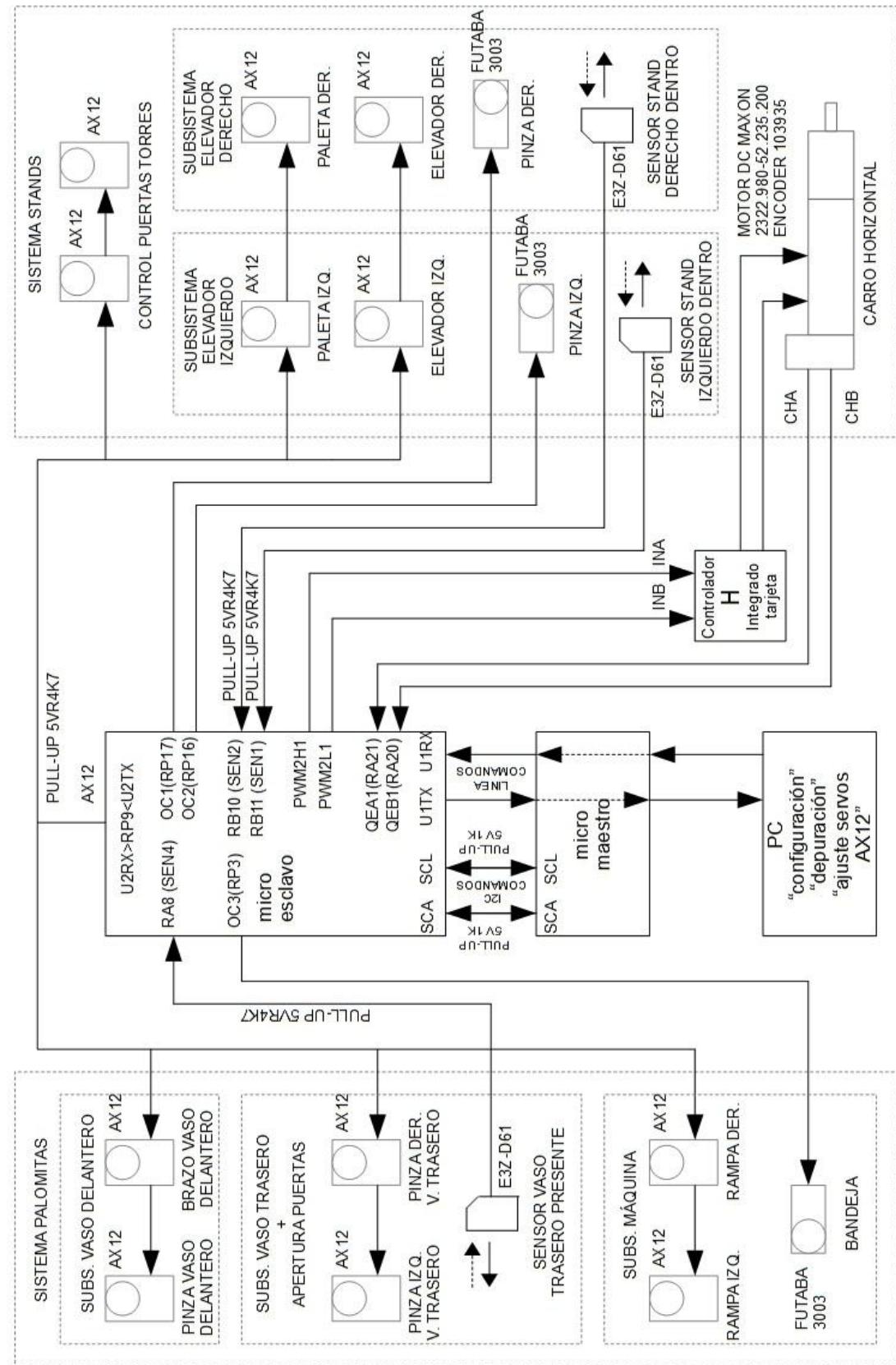


Figura 2-1 Esquema conexiones de micro al resto de elementos del sistema

En la Tabla 2-1 *Características de conexiones a pines de microcontrolador utilizados* Tabla 2-1 se definen las especificaciones detalladas de cada una de los pines utilizados.

PIN/ FUNCIO N	I/O	TIPO	FUNCIÓN	Descripción	Actuador
RP9	I		RX	Recepción status AX12 half-duplex	12 Servos Dynamixel AX12
	O	Colector abierto	TX	Transmisión comando AX12 half-duplex	
RP17	O		PWM	Pinza elevador derecho	Servo Futaba 3003
RP16	O		PWM	Pinza elevador izquierdo	Servo Futaba 3003
RP3	O		PWM	Apertura de bandeja	Servo Futaba 3003
RA8	I	Sensor colector abierto + Pull-up	Entrada digital	Sensor vaso trasero presente (ON)	Sensor reflexivo difuso E3Z-D61 NPN
RB10	I	Sensor colector abierto + Pull-up	Entrada digital	Sensor stand derecho dentro (ON)	Sensor reflexivo difuso E3Z-D61 NPN
RB11	I	Sensor colector abierto + Pull-up	Entrada digital	Sensor stand izquierdo dentro (ON)	Sensor reflexivo difuso E3Z-D61 NPN
PWM2H 1	O		INA	Control lado A puente en H	Driver + puente H integrado en tarjeta
PWM2L 1	O		INB	Control lado B puente en H	
RP20	I		CHB	Entrada en cuadratura canal B encóder	Maxon sensor ENC 20, 100CPT, 2 CANALES
RP21	I		CHA	Entrada en cuadratura canal A encóder	
RP7	O		TX	Transmisión UART PC	PC a través de maestro
RP8	I		RX	Recepción UART PC	
RB5	I/O	Salida colector abierto	SDA	Entrada salida de datos de módulo I2c	Micro maestro
RB6	I/O	Salida colector abierto	SCL	Entrada-salida de reloj de módulo I2c	Micro maestro

Tabla 2-1 *Características de conexiones a pines de microcontrolador utilizados*

## 2.2. Mapeo de entradas-salidas

En el desarrollo de la tarjeta de control por parte de la asociación de robótica de Coslada se hizo uso para el interconexionado de la funcionalidad que ofrecen los dsPIC para el remapeado de pines. Esta funcionalidad permite redireccionar por software la salida de los módulos a una serie de pines que permiten el remapeo.

En la Figura 3-1 se esquematiza el conexionado de los diferentes elementos del sistema a los pines físicos del micro y en caso de ser necesario el remapeado de estos pines a los módulos de control.

En la Tabla 2-2 se detallan los pines con necesidad de remapeo por módulos.

PIN/ FUNCION	I/O	PIN FUNCIÓN	MÓDULO
RP9	I	U2RX	UART 2
	O	U2TX	
RP17	O	OC1	MÓDULO SALIDAS COMPARACION
RP16	O	OC2	
RP3	O	OC3	
RP20	I	QEB1	MÓDULO INTERFACE ENCODER CUADRATURA
RP21	I	QEA1	
RP7	O	U1TX	UART 1
RP8	I	U1RX	

Tabla 2-2 Pines de entrada-salida de módulos periféricos de microcontrolador remapeado

Tanto la salida TX como la entrada RX de la UART2 tienen que ser remapeadas al pin RP9 para simular el funcionamiento del protocolo half-duplex de control de los servos AX12. La explicación detallada del protocolo se encuentra en el apartado 5.1.11. Esta solución que evita el empleo de hardware adicional tiene el inconveniente de que complica la gestión de las operaciones de lectura y escritura, debido a que todo lo que es transmitido por el pin TX es recibido por el RX al mismo tiempo. Por lo tanto es necesario desechar toda esta entrada de datos.

### 3. Manual de implementación SW detallado

#### 3.1. Arquitectura SW detallada

En el apartado 5.2. “especificación de protocolo de comunicación con microcontrolador maestro” ya se ha definido la estructura de datos que será enviada desde el micro maestro como comando y la estructura de estado que enviará el esclavo bajo petición del micro maestro. El canal para esta comunicación será el bus I2C que conecta el micro maestro con el esclavo y la tarjeta de expansión de entradas del maestro.

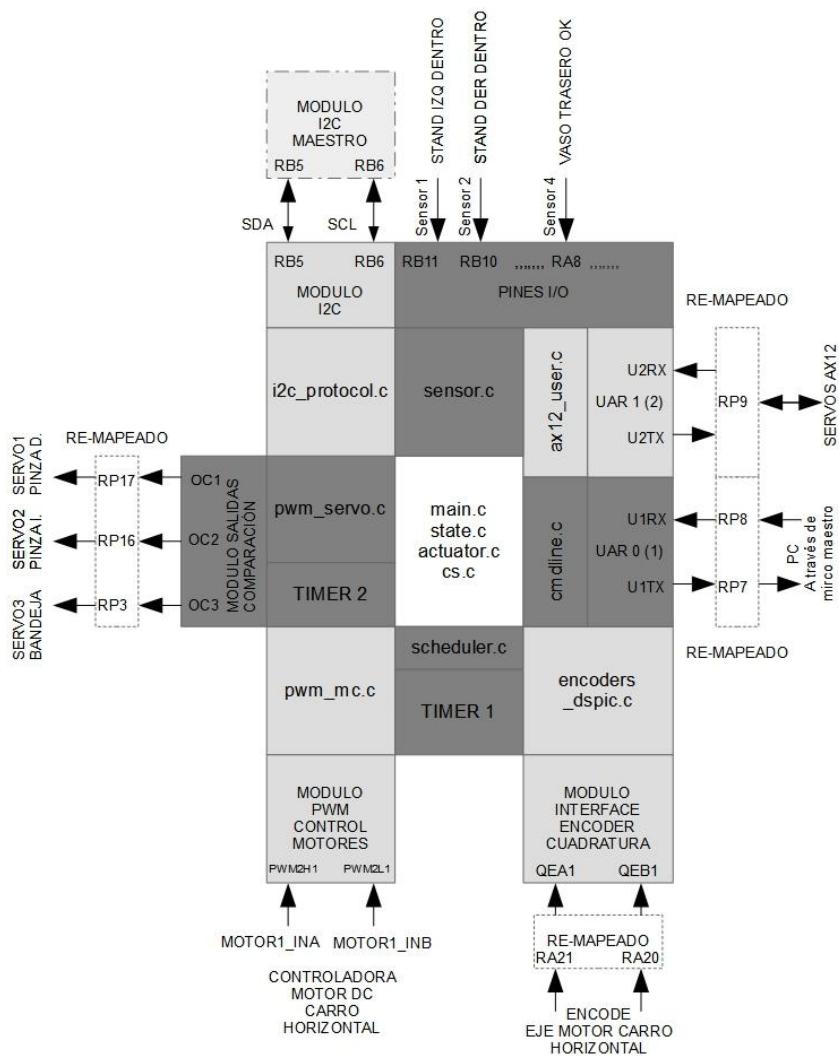


Figura 3-1 Esquema conexiones a módulos periféricos de microcontrolador con etapas de remapeado de pines

Por otro lado se establecerá un segundo canal de comunicación que se empleara para depuración y configuración desde la consola del PC, a través de la UART. Para esto se utilizará el módulo interfaz hombre-máquina IHM de Aversive, ampliando su funcionalidad con la codificación de los mismos comandos que pueden ser recibidos en la estructura de comando maestro por I2c. El funcionamiento para las órdenes de entrada por ambos canales debe ser el mismo.

Se han codificado 5 tipos de comandos fundamentales que deberán ejecutarse según lo expuesto, por ambos canales:

- Inicialización actuadores por color de campo
- Power OFF
- Actuadores simples
- Comandos de depuración sobre un único actuador que pueden bloquear ejecución hasta conclusión de movimiento en servos AX12.
- Actuadores múltiples
- Comandos de juego que no deben ser bloqueantes. Tienen que permitir un cambio de comando en cualquier momento.
- Modificación variables de estado

---

### **3.2. Control servos AX12**

Una vez iniciado un movimiento en un servo AX12 se ha controlado periódicamente la posición hasta que alcance la posición de destino con un umbral de incertidumbre de 6 grados. Se ha limitado el tiempo máximo de espera de final de movimiento al triple del tiempo esperado para el movimiento. No se ha establecido limitación de velocidad ni de torque. El tiempo de movimiento se ha calculado en función a la velocidad máxima del servo en vacío 300 grados/segundo (3333 useg/grado) y de los grados a recorrer. En caso de time-out se informa del estado de bloqueo de sistema.

---

### **3.3. Comandos uart PC de depuración y configuración**

Ha sido necesario codificar la lista completa de comandos que pueden ser enviados desde el microcontrolador maestro. Esto incluye la generación de las estructuras de datos de módulo IHM y la función de comando recibido por cada instrucción. En estas funciones se ha simulado la recepción de un comando por I2c generando la misma reacción.

La lista completa de comandos es la siguiente. Siendo el formato que se ha empleado para mostrar la estructura de cada comando con sus diferentes modificadores:

[nombre comando] + [argumento 1] + [argumento 2] + [argumento n]

Siendo:

[argumento 1] = [modificador 1#modificador 2#modificador N] o [VALOR]

[argumento 2] = [modificador 1#modificador 2#modificador N] o [VALOR]

[argumento N] = [modificador 1#modificador 2#modificador N] o [VALOR]

- Comandos (escritura i2c) de inicialización de actuadores, power off y lectura de estructura de estado (lectura i2c)
   
INIT\_LEFT-INIT\_RIGHT-POWER\_OFF-STATUS
   
[state] [init\_left#init\_right#power\_off#status]
- Comando de control de posición de ángulo de paletas de stands.
   
STANDS\_BLADE (lado+posición absoluta+offset)
   
[blade\_left] [hide\_left#push\_left#center#push\_right#hide\_right#angle]
   
[OFFSET]
   
[blade\_right] [hide\_left#push\_left#center#push\_right#hide\_right#angle]
   
[OFFSET]
- Comando de control de posición de ángulo de apertura de las pinzas de los elevadores de stands.
   
STANDS\_CLAMP (lado+posición absoluta+offset)
   
[clamp\_left] [full\_open#open#close] [OFFSET]
   
[clamp\_right] [full\_open#open#close] [OFFSET]
- Comando de control de posición de los elevadores de stands.
   
STANDS\_ELEVATOR (lado+posición absoluta+offset)
   
[elevator\_left] [up#down] [OFFSET]
   
[elevator\_right] [up#down] [OFFSET]
- Comando de control de apertura de las puertas de las torres de stands.
   
STANDS\_TOWER\_CLAMP (posición absoluta+offset)
   
[tower\_clamps] [unlock\_left#lock#unlock\_right] [OFFSET]
- Comando de control de posición de ángulo de las pinzas del vaso trasero. La posición de apertura máxima produce indirectamente la apertura de las puertas de descarga de las palomitas.
   
CUP\_CLAMP\_POPCORN\_DOOR (lado+posición absoluta+offset)
   
[clamp\_door\_left] [hide#cup\_locked#cup\_unlocked#door\_close#door\_open]
   
[OFFSET]
   
[clamp\_door\_right] [hide#cup\_locked#cup\_unlocked#door\_close#door\_open]
   
[OFFSET]
- Comando de control de apertura de la bandeja de las palomitas.
   
POPCORN\_TRAY (posición absoluta+offset)
   
[tray] [open#close] [OFFSET]

- Comando de control de apertura de las rampas de las palomitas.  
POPCORN\_RAMPS (posición absoluta+offset)  
[ramps] [hide#harvest#open] [OFFSET]
- Comando de control de posición de ángulo de apertura de la pinza del vaso delantero.  
CUP\_CLAMP\_FRONT (posición absoluta+offset)  
[cup\_clamp\_front] [open#cup\_locked] [OFFSET]
- Comando de control de posición de ángulo del brazo de la pinza del vaso delantero.  
CUP HOLDER\_FRONT (posición absoluta+offset)  
[cup\_holder\_front] [cup\_hold#ready#hide] [OFFSET]
- Comando de control sobre el sistema de la máquina de las palomitas.  
POPCORN\_SYSTEM (comando)  
[ps][idle#cup\_f\_ready#cup\_f\_drop#cup\_f\_release#cup\_f\_hide#cup\_r\_open#cup\_r\_catch  
#cup\_r\_release#machines\_ready#machines\_harvest#machines\_end#stock\_drop#  
stock\_end]
- Comando de control sobre las dos torres de almacenamiento de stands.  
STANDS\_SYSTEM (lado+comando+angulo paleta)  
[ss] [left#right]  
[idle#hide\_tower#harvest\_stand\_do#harvest\_stand\_ready#build\_spotlight  
#release\_spotlight] [ANGULO\_PALETA]
- Comando de ajuste el valor de la variable de estado al indicado.  
SET\_INFOS (cup\_front\_catched, cup\_front\_catched, cup\_rear\_catched,  
machine\_popcorns\_catched, stored\_stands\_l, stored\_stands\_r)  
[set\_infos]  
[cup\_front\_catched#cup\_rear\_catched#machines\_catched#stands\_left#stands\_right]  
[VALOR DE AJUSTE]

### **3.4. Estructura de datos**

---

Las estructuras de datos creadas para el control de los sistemas han heredado la metodología utilizada en las librerías Aversive. En estas librerías todos los datos de configuración y estado de un módulo han sido agrupados en una misma estructura de datos. Esta incluye solo aquellos datos que son de ámbito de uso externo a las funciones del módulo, y es pasada como parámetro a las funciones del alto nivel de los módulos cuando son invocadas desde el exterior del módulo.

En algunos casos estas estructuras contienen punteros a las funciones de los submódulos que han sido configurados para su ejecución dentro del módulo. Un ejemplo de esta situación es el módulo de control de bucle “cs”. Dispone de una estructura donde están almacenadas las direcciones de las funciones de los filtros que han sido elegidos para su ejecución dentro del esquema del bucle y en este caso especial también punteros a las estructuras de configuración de los filtros y por último las variables de estado del sistema. Las estructuras de filtro también dirigen la función de ejecución de filtro.

Este sistema de datos permite la definición de una estructura por cada sistema a controlar, dependiendo el número máximo, de la capacidad del módulo hardware asociado al sistema.

El encadenamiento de estructuras se transmite desde el bajo nivel hasta el más alto nivel, estando en el último escalafón dos estructuras que almacenan todas las demás estructuras según sean de sistemas genéricos o asociadas a la programación específica de la aplicación. Estas son respectivamente la estructura “genboard” y la estructura “slavedspic”, esta última para nuestra aplicación en particular.

En la Figura 3-2 se muestra el árbol de anidamiento de estructuras, destacando en color rojo las estructuras que han sido creadas para la aplicación.

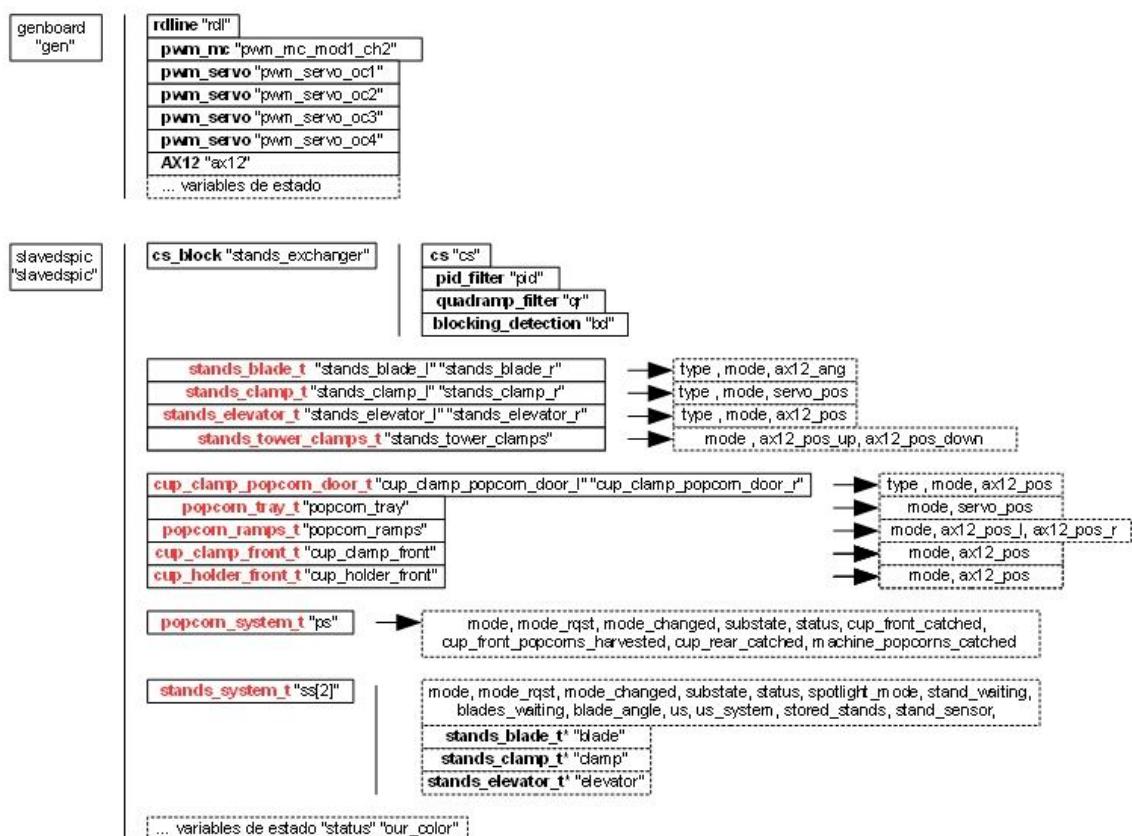


Figura 3-2 Esquema de anidamiento de estructuras de datos del proyecto

Los recuadros en cursiva corresponden a las variables que contienen las estructuras.

Se ha definido una estructura por cada actuador o grupo de dos actuadores de dos sistemas simétricos con funcionamiento independiente, o grupo de dos actuadores con funcionamiento siempre simultáneo de forma simétrica o asimétrica, declarándose una por lado para cada actuador de sistemas simétricos independientes. El último de los tres casos se da con “stands\_tower\_clamps\_t” y “popcorn\_rampe\_t” en los que el sistema formado por dos actuadores con movimiento combinado es almacenado en una misma estructura. En estos casos se han definido dentro de la estructura una variable de offset por cada uno de los actuadores. En el resto de cosas la estructura solo contiene type (lado en sistema simétricos), mode (posición de actuador codificada) y posición absoluta u offset en pulsos “\_pos” o ángulo “\_ang” según el mode.

Estas variables almacenan la posición programada en el actuador correspondiente, estando en el caso de los analógicos ajustada la PWM o de los digitales almacenada la posición en el registro de interno del AX12. Por lo tanto es solo la orden de posición enviada a los actuadores y no corresponde a la alcanzada.

Estas estructuras de datos junto con la estructura de comando “i2c\_cmd\_slavedspic\_set\_mode” de ajuste de modo de esclavo por comando i2c conforman toda la estructura de datos del sistema.

---

### 3.5. Estructura de programa

---

El código creado ha sido estructurado en 3 niveles de abstracción diferentes con toda la base de las Aversive por debajo de ellos. Estos niveles se encuentran claramente separados en tres archivos, con sus correspondientes archivos de cabecera asociados. En el diagrama 6 se ha esquematizado de forma simplificada la interdependencia entre estos tres archivos “main.c”, “state.c” y “actuator.c”, y el resto de archivos de Aversive de los que se hace uso en la aplicación. Aunque se han omitido los archivos de cabecera, si se han incluido los archivos de configuración de módulos.

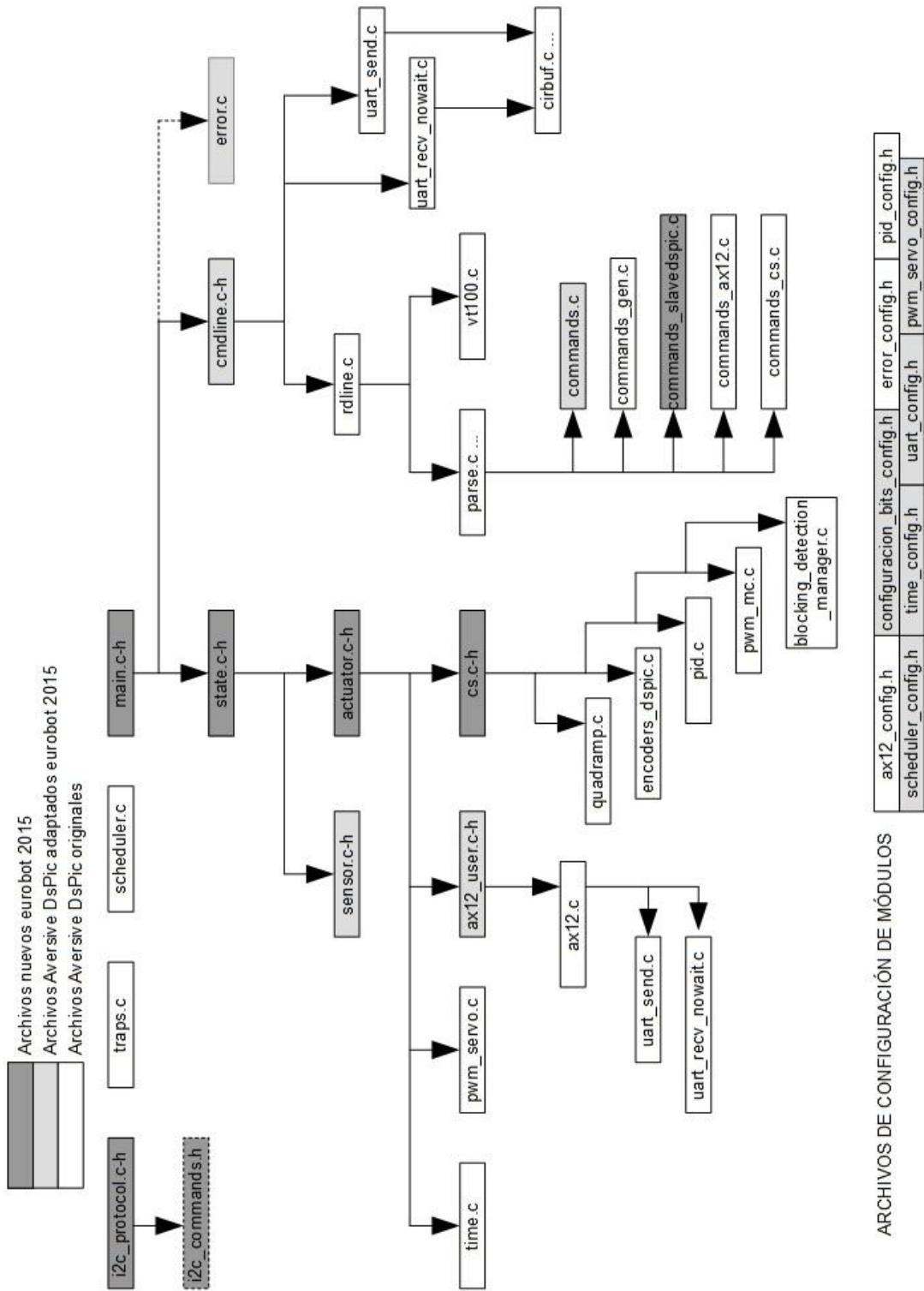


Diagrama 3-1 Diagrama de interdependencias entre archivos de código del proyecto

### **3.5.1. Archivo “main.c”**

---

Este archivo corresponde por un lado al mayor nivel de abstracción contenido la función main( ) y desde esta la llamada a las funciones de más alto nivel y por otro el mayor acercamiento al más bajo nivel con la inicialización y configuración global del sistema alcanzando los pines del microcontrolador.

La función main está dividida en dos partes. Una inicial de configuración en la que es inicializado el sistema íntegramente y una final con un bucle de ejecución infinita en la que se llama a las funciones de los dos sistemas que van a ejecutarse de manera paralela indefinidamente. Estos sistemas van a ser la máquina de estados de control del sistema y el interfaz IHM para la comunicación por la línea de comandos desde el PC.

En este archivo se han creado varias funciones de configuración que son invocadas en la zona de inicialización inicial de main y cuya creación tan solo atiende a facilitar la comprensión del código. Con la excepción de la función de atención a la interrupción del Timer 1.

Estas funciones son:

- **void io\_pins\_init(void)**  
Configuración de los pines del micro como entrada o salida a través de los registros TRIS.  
Configuraciones de pines de salida en configuración de colector abierto a través de registros ODC.  
Inicialización de la salida a través del LAT en pines configurados como salida.  
Re-mapeado de pines de módulos en otros pines del micro a través de los registros de funciones especiales.
- **void timer\_init(void)**  
Mediante esta función se configura el Timer 1 para operar como base de tiempo para la ejecución periódica de la función de gestión del módulo de tareas periódicas programables “scheduler”.  
Es imprescindible la configuración manual al no estar definida la macro “CONFIG\_MODULE\_SCHEDULER\_TIMER1” que habilita la configuración automática desde la librería del scheduler. Esta configuración será detallada en el apartado 5.6.2. “Gestión de scheduler. Tareas de ejecución periódica”.
- **void \_\_attribute\_\_((\_\_interrupt\_\_, no\_auto\_psv)) \_T1Interrupt(void)**  
Función de atención a la interrupción del Timer 1. En esta función tan solo se baja el flag de interrupción y se invoca a la función de gestión del Scheduler.

### **3.5.2. Archivo “state.c”**

---

En este archivo se encuentra la función de control del sistema “state\_machines()“ que es invocada desde el bucle infinito de main.

El resto del contenido está dividido en cinco secciones claramente diferenciadas:

- **Función de cambio de comando. state\_set\_mode(struct i2c\_cmd\_slavedspic\_set\_mode \*cmd)**

Esta función gestiona los nuevos comandos recibidos. Es invocada por un lado desde las funciones de comando recibido por la línea de comandos UART desde el PC y por otro desde el módulo I2c de control de nuevo comando recibido. Recibe la referencia de una estructura de comando `i2c_cmd_slavedspic_set_mode*` e internamente carga está en la estructura de comando del sistema `mainboard_comman` y sube los flag de nuevo comando.

- **Funciones de ejecución de comandos de actuadores simples**

Estas funciones son invocadas desde la función de máquinas de estado y existe una por cada comando de actuador simple. Su uso es exclusivo para depuración y tienen una ejecución bloqueante, no retornan hasta que no se ha concluido el movimiento del actuador o se ha concluido que el movimiento ha sido bloqueado. En cada iteración del bucle de main dentro de cada una de estas funciones se comprueba si el comando actual es el correspondiente a esta función, y en cuyo caso si además está activo el flag de nuevo comando se ejecuta el comando correspondiente.

- **Funciones de gestión de sistemas de actuadores múltiples**

Estas funciones son invocadas desde la función de máquinas de estado y existe una para el sistema de palomitas y otra para los dos sistemas de stands paralelos. Gestionan la ejecución de la función del comando activo del sistema, siendo cada una de estas una máquina de estado. Al mismo tiempo gestionan la variable de estado asociada a cada sistema según se encuentre sin comando (en espera de comando), hecho (estado previo a sin comando), error, etc. Son de ejecución no bloqueante con una estructura típica de múltiples máquinas de estado. En el caso de la función de los sistemas de stands está preparada para ser invocada por cada uno de los dos sistemas, el derecho y el izquierdo, recibiendo como parámetros la estructura del sistema que actúa como maestro y la del que actúa como esclavo. Tan solo será utilizada la estructura del esclavo para el comando de construcción de foco que sincroniza los movimientos de ambos sistemas.

- **Funciones de ejecución de comandos de actuadores múltiples**

Estas funciones son invocadas desde las funciones de gestión de sistemas de actuadores múltiples. Cada una es una máquina de estado de la secuencia de movimientos solicitada.

- **Funciones de ejecución de comandos especiales**

Dentro de estas funciones se encuentran las siguientes:

- Función de comandos de inicialización del sistema de actuadores por campo “state\_init( )”

Esta función es invocada desde la función de máquinas de estado y atiende a los comandos de inicialización por juego en campo izquierdo y por campo derecho

- Función de comando de ajuste de variables de estado “state\_do\_set\_infos( )”  
Esta función es invocada desde la función de máquinas de estado y atiende al comando “SET\_INFOS”.

A continuación se van a detallar las estructuras tipo, de cada grupo de funciones, que salvo diferencias puntuales mantienen el mismo sistema de ejecución.

En el Diagrama 3-2 se muestra el diagrama de flujo de la función de control del sistema “state\_machines( )”. Desde esta función se llama por orden a cada una de las funciones de ejecución de comandos, que por orden son:

- Funciones de ejecución de comandos especiales (bloqueante)
- Funciones de ejecución de comandos de actuadores simples (bloqueante)
- Funciones de gestión de sistemas de actuadores múltiples (maquinas de estado no bloqueantes).

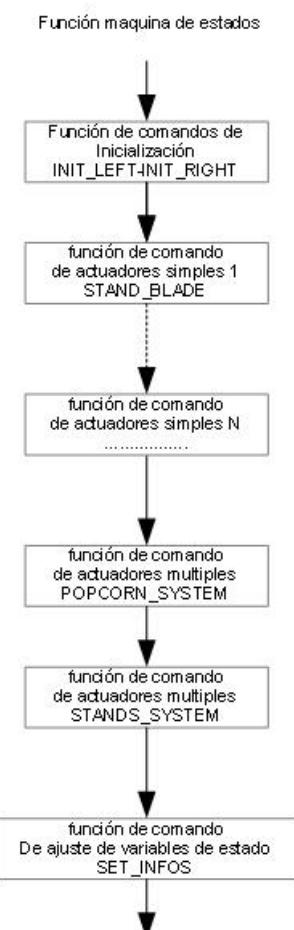


Diagrama 3-2 *Diagrama de flujo de función de máquina de estados principal*

### **3.5.3. Estructura funciones de ejecución de comandos de actuadores simples**

Estas funciones controlan una orden de movimiento simple de un sistema formado por uno o dos actuadores que funcionan con movimientos simétricos. No reciben ni retornan ningún parámetro, sino que operan directamente con la estructura de comando solicitado “mainboard\_command” y la variable global “mode\_changed” que actúa de flag de nuevo comando solicitado.

Cuando son ejecutadas comprueban si se ha solicitado un nuevo movimiento del actuador comprobando que el comando activo sea el correspondiente al del actuador(es) y este activo el flag de nuevo comando. En caso negativo se retornara directamente de la función y en caso positivo se bajara el flag de nuevo comando y se procederá al movimiento del actuador.

No se retornara de la función hasta que no se haya finalizado el movimiento o finalice la temporización de tiempo máximo para el movimiento. Este tiempo es gestionado por las funciones de más bajo nivel que calculan el tiempo estimado para el movimiento en función de la velocidad máxima del servo y los grados de desplazamiento programados.

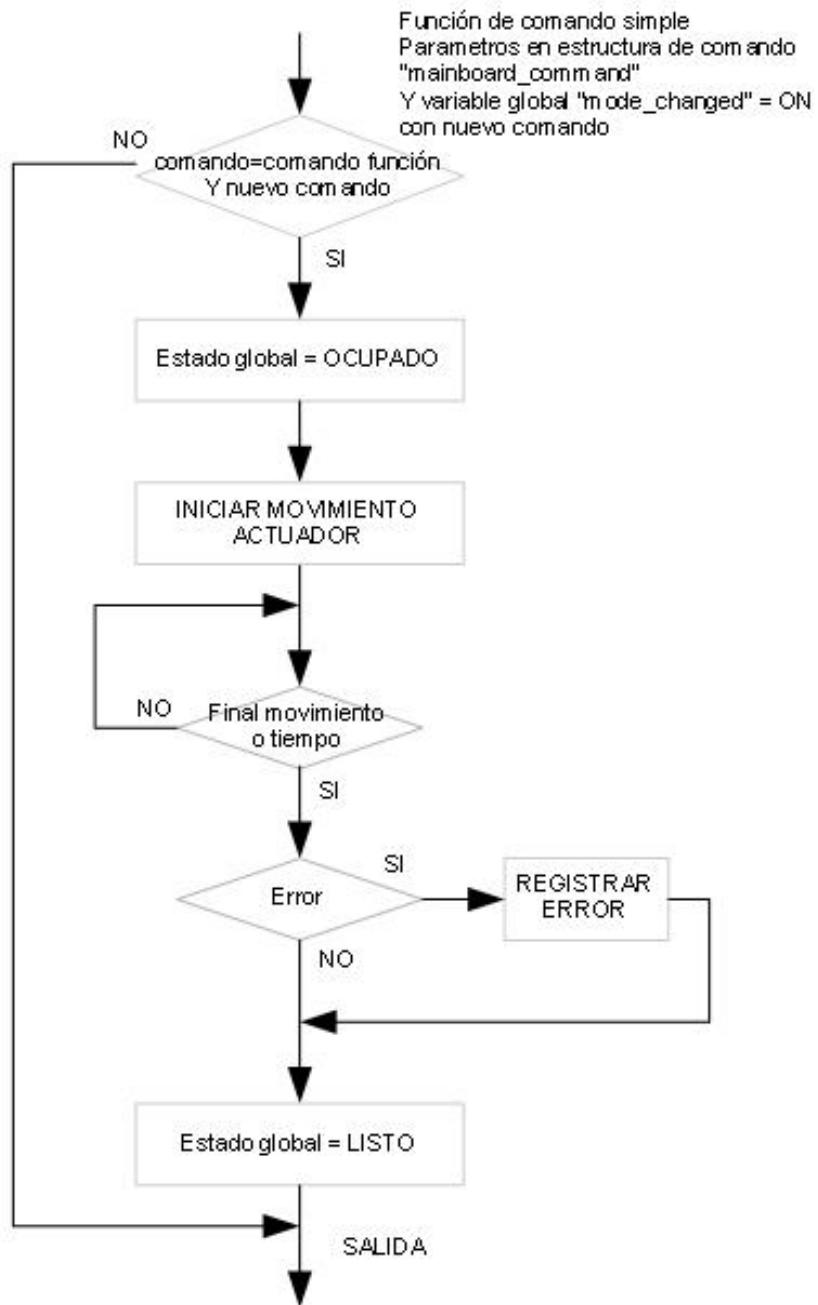


Diagrama 3-3 Diagrama de flujo “tipo” de gestión de actuadores simples

### **3.5.4. Estructura funciones de gestión de sistemas de actuadores múltiples**

---

Estas funciones también mantienen la misma estructura independientemente del sistema a controlar. Se puede ver el diagrama de flujo general en el Diagrama 3-4.

Reciben como parámetros en el caso del sistema de palomitas la estructura de datos del sistema y en el caso de los sistemas paralelos de stands (las dos torres de elevación junto con los sistemas asociados) la estructura del lado que funciona como maestro y la el lado que operara como esclavo para el comando de construcción de foco.

Inicialmente comprueban la variable contenida en la estructura de datos del sistema de flag de nuevo comando y en caso de estar activa actualizan el comando en ejecución al valor del comando solicitado y se inicializa la variable de la máquina de estados del estado inicial.

A continuación establecen una frecuencia de ejecución cada 5 milisegundos, retornando de la función si no se ha alcanzado ese tiempo desde la última ejecución.

Como siguiente paso se comprueba si existe un comando en ejecución, y se ejecuta la función de la máquina de estados, del comando correspondiente. La máquina de estados ejecuta una secuencia de movimientos, comprobando la finalización de cada uno de ellos.

Cuando se retorna de la función de la máquina de estados se comprueba el estado de ejecución devuelto por esta y almacenado en la variable de estado del sistema. Si el estado es hecho “STATUS\_DONE”, la función del comando informa que se acaba de concluir la ejecución de la secuencia de movimientos y se pasa a comprobar si el comando esta constituido por una agrupación de varios comandos mas sencillos. En caso afirmativo se actualiza el comando en ejecución al valor del comando siguiente de la secuencia y en caso negativo se cambia el estado del comando activo a sin comando activo. Este sistema de comandos compuestos se ha creado para reutilizar secuencias de movimientos comunes en diferentes comandos.

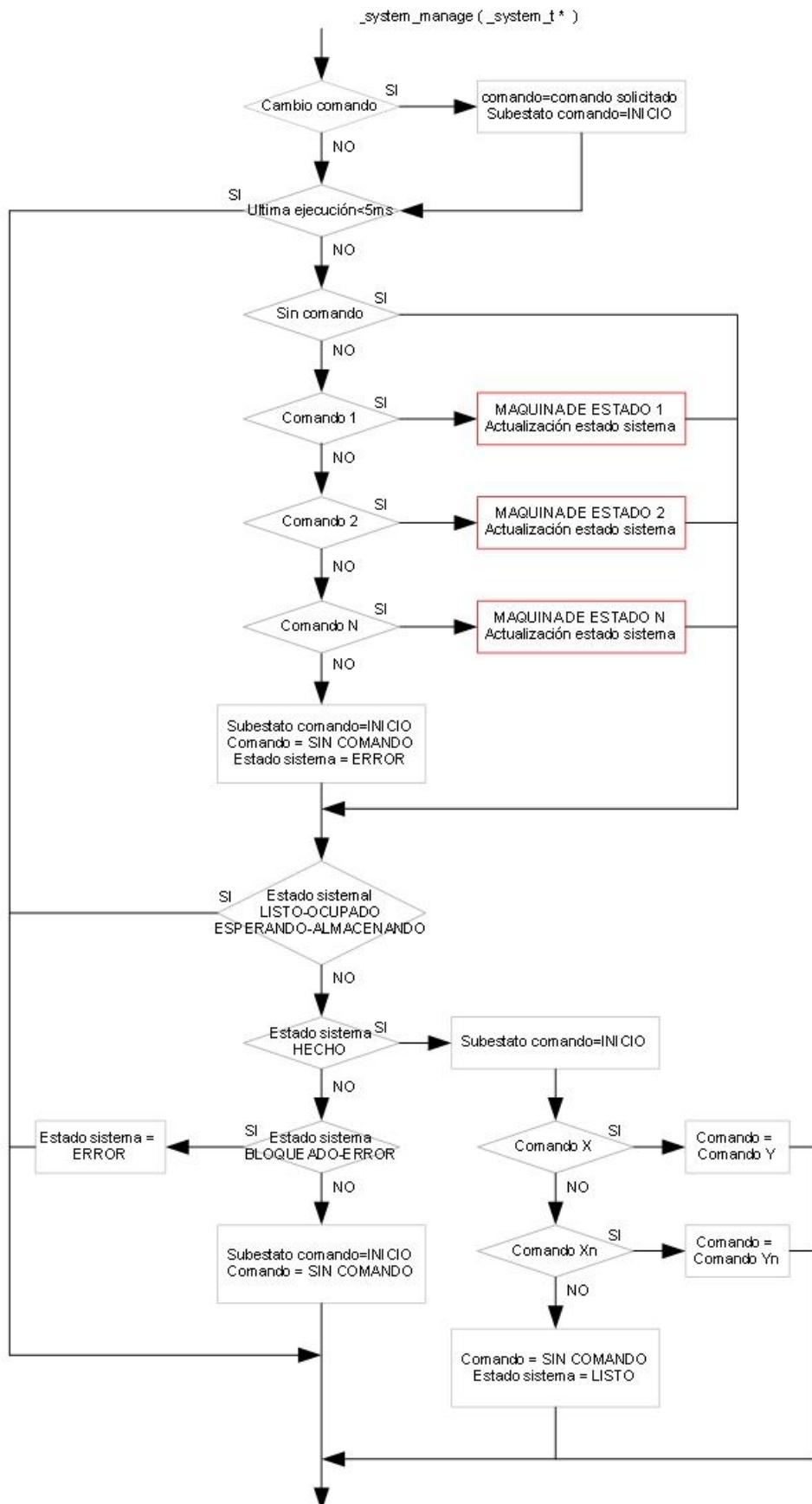


Diagrama 3-4 Diagrama de flujo “tipo” de funciones de gestión de actuadores múltiples

### **3.5.5. Estructura funciones de ejecución de comandos de actuadores multiples**

Estas funciones también mantienen la misma estructura base independientemente de la secuencia de movimientos a realizarse.

Reciben como parámetros en el caso de las funciones del sistema de palomitas la estructura de datos del sistema, y en el caso de las de los sistemas paralelos de stands, también la estructura de datos del sistema con la excepción de la función de construcción de foco, que recibe la estructura del lado que opera como maestro y la el lado que operara como esclavo. Esta función también es una excepción en lo referente a su estructura ya que llama a dos funciones que controlan cada uno de los sistemas (torres paralelas) con unas señales de sincronización entre ambas.

En el caso general la estructura está formada por una sencilla máquina de estados que ejecuta en cada iteración solo el estado actualmente activo, se puede ver el diagrama de flujo general en el Diagrama 3-5.

Existen tres tipos fundamentales de estado:

- De iniciación de movimiento, en el que se actúa en un actuador mediante la función asociado y se cambia inmediatamente al próximo estado para su ejecución desde la próxima iteración. Este estado salvo pocas excepciones será un estado de espera de finalización de movimiento.
- De espera de finalización de movimiento en servos digitales AX12, en el que no se cambiara al próximo estado hasta que no se concluya el movimiento programado en el actuador o se sobrepase el tiempo máximo de ejecución. En este último la función retornara el estado de bloqueo.
- De espera de finalización de movimiento en servos analógicos PWM, en el que no se cambiara al próximo estado hasta que no se concluya la temporización fijada para el movimiento.

Una vez finalizado el último estado informa a la función desde la que es invocado retornando el estado hecho.

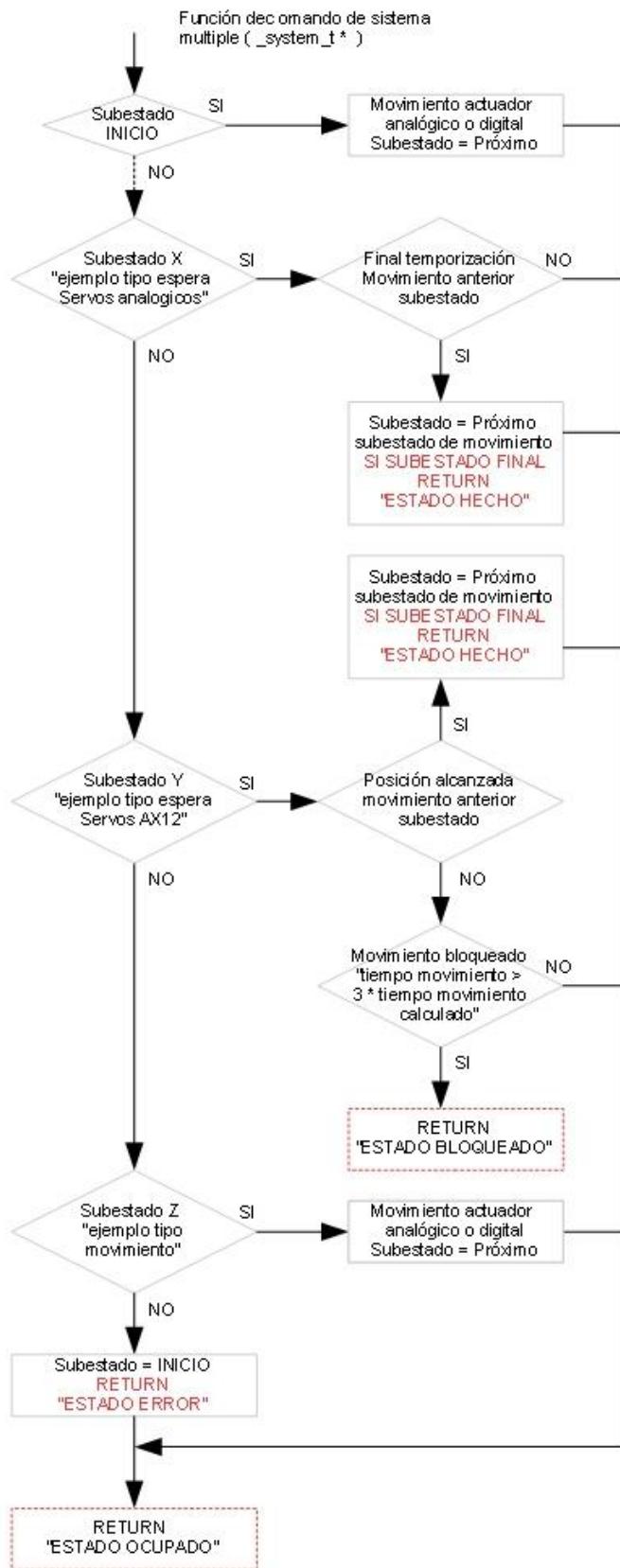


Diagrama 3-5 Diagrama de flujo “tipo” de funciones de ejecución de actuadores múltiples

### 3.5.6. Archivo “actuator.c”

---

En este archivo se han agrupado funciones con dos diferentes niveles de abstracción. Por un lado todas las funciones que son invocadas para el control de los diferentes actuadores desde las funciones del archivo “state.c”, y por otro las funciones de más bajo nivel de control de los servos AX12 y el motor del carro horizontal y que son a su vez invocadas desde las funciones de control de los diferentes actuadores de este mismo archivo.

Para las funciones de control de los actuadores se ha creado una función por cada actuador o grupo de dos actuadores de dos sistemas simétricos con funcionamiento independiente, o grupo de dos actuadores con funcionamiento siempre simultáneo de forma simétrica o asimétrica. Como ejemplo de cada uno de ellos se pueden mencionar los siguientes por orden de mención:

- El servo analógico que controla la bandeja de las palomitas
- Los dos servos AX12 que controlan el movimiento de elevación de cada uno de los almacenes de stands.
- Los dos servos AX12 de las rampas de las palomitas.

Existe una función por cada una de las estructuras de datos que se crearon para los diferentes actuadores.

Estas funciones van a recibir como parámetros un puntero a la estructura de datos del actuador a controlar, la nueva posición solicitada y el offset sobre esta. Existe un caso especial según las especificaciones en el que en la posición se indica que el offset será utilizado como valor absoluto de desplazamiento.

Para los actuadores formados por servos AX12 se han creado dos funciones más por actuador o grupo de actuadores según lo anterior. Ambas funciones controlan la finalización del movimiento programado, una bloqueando la ejecución hasta que sea finalizada o concluya el tiempo para el movimiento y la otra comprobando sin bloquear si concluyó o no dicho movimiento.

### 3.5.7. Estructura funciones de control de actuadores

---

Asociado a cada una de estas funciones se han declarado una serie de arrays de una o dos dimensiones, según se trate de un solo actuador o agrupaciones de 2 actuadores, en los que se han almacenado el valor de ajuste del registro asociado a la posición de cada actuador, para cada uno de los diferentes posicionamientos que pueden ser solicitados al actuador mediante los comandos de mando.

Estas funciones reciben junto con la estructura de datos de cada actuador o agrupación, el nuevo posicionamiento y el offset sobre este. Se puede ver el diagrama de flujo general en el diagrama 11.

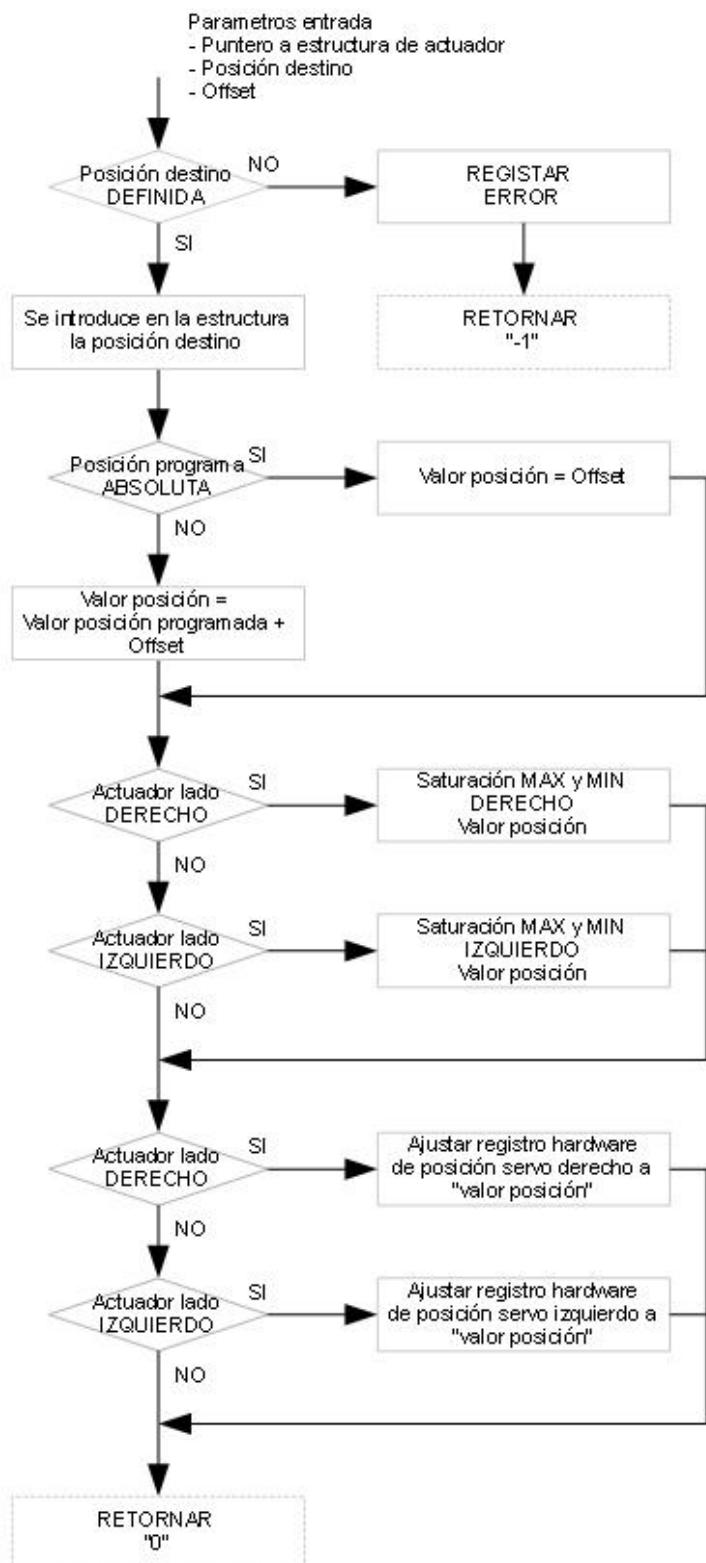


Diagrama 3-6 Diagrama de flujo "tipo" de funciones de control de actuadores

Explicación etapas diagrama de flujo:

- Se comprueba que el código del posicionamiento solicitado esta dentro del rango de los codificados para este actuador. En caso negativo se sale de la función retornando el código de error “-1” y registrando un error.
- Se carga el código de posicionamiento en la variable de código de posición en la estructura de datos del actuador o grupo de actuadores.
- Si el código cargado, corresponde con el de posición absoluta pasada como offset, se carga el valor del offset en la variable de la estructura de datos de posición en pulsos o grados según se haya codificado para cada actuador, en caso contrario se carga el valor en pulsos o mm correspondiente a la posición solicitada y que como se ha explicado se encuentra almacenada en el array asociado y se le suma el valor del offset solicitado y este si pasado en las unidades de control del actuador (pulsos o mm).
- Para los casos en los que el sistema esta formado por dos actuadores con funcionamiento simétrico independiente, se tiene en cuenta si la estructura pasada como parámetro, corresponde al lado derecho o al izquierdo, para las dos siguientes etapas.
- Saturación del valor de posición calculada en pulsos o mm a los valores máximo y mínimos del lado correspondiente al actuador. Es necesario esta comprobación ya que el valor del offset aplicado no se controla hasta este punto.
- Se carga el valor finalmente calculado de posición en pulsos o mm al registro del actuador, mediante las funciones de bajo nivel de control de los actuadores. Para el caso de los servos AX12 existe una función para programar la posición destino en pulsos absolutos o en grados. En el caso de los grados el centro del rango corresponde con los 0 grados.
- Si se finaliza la función sin ningún error esta retorna devolviendo “0”, que indica que la ejecución fue correcta.

### **3.6. Líneas de ejecución**

En este apartado se van a describir los diferentes caminos de ejecución de la aplicación a través de todos los sistemas que tienen habilitadas interrupciones.

En el Diagrama 3-7 se han esquematizado todas las líneas de ejecución indicando la fuente de interrupción de los diferentes módulos que las tienen habilitadas y las funciones y sistemas que hacen uso de ellas.

Como ya se ha explicado, de la función main cuelgan dos funciones principales de ejecución continua. Respecto a las características de ejecución, de la función de máquina de estados, es importante destacar que contiene zonas de ejecución periódica cada 5 milisegundos.

- El timer 1 genera una interrupción periódica de 512 microsegundos, en la que se ejecuta la función de gestión de tareas periódicas programables Scheduler. Esta función ejecuta las siguientes funciones:

- Cada 2 ms la función de gestión de los módulos asociados al control de bucle del motor de carro horizontal. Dentro de estos módulos está el de lectura del encóder, control de bucle PID de motor y control de bloqueo de motor.
  - Cada 2,5 milisegundos la función “time\_increment( )” que incrementa el contador de tiempo absoluto del sistema.
  - Cada 10 milisegundos la función “do\_sensor( )” actualiza y filtra todas las variables de estado de las entradas digitales.
  - Cada 8 milisegundos la función “do\_i2c\_wachdog( )” comprueba que desde la última ejecución se haya recibido desde el micro maestro por i2c una petición de lectura de la estructura de variables de estado del sistema.
- En la UART 0 que es la encargada de las comunicaciones del módulo IHM de línea de comandos están habilitadas como fuentes de interrupción el flag de registro de salida de TX vacío y el flag de nuevo carácter recibido en RX.
    - Cuando el registro de salida está vacío se llama a la función “uart\_send\_next\_char( )” que comprueba si hay datos sin enviar en el buffer fifo de salida asociado al módulo IHM y en caso afirmativo envía el siguiente dato.
    - Cuando se recibe un nuevo dato en el registro de entrada de la uart se ejecutan dos funciones diferentes:
      - Función “uart\_recv\_next\_char( )”. Asociado al módulo IHM de linea de comandos por la uart 0. Almacena el byte recibido en el buffer fifo de entrada asociado al módulo IHM.
      - Función “emergency( )”. Esta función recibe el carácter recibido por la uart de forma paralela a la función anterior y comprueba la recepción consecutiva de los caracteres “p” “o” “p”. En caso de sucederse esta secuencia resetea el robot.
- En la UART 1 que es el canal de comunicación con los servos AX12 está habilitada como fuente de interrupción el flag de registro de salida de TX vacío. Esta interrupción ejecuta la función “ax12\_send\_callback( )” que cambia el pin donde están mapeadas tanto la salida TX como la entrada RX de la uart 1 de salida a entrada. En este pin se simula una comunicación half dúplex y cuando se termina una transmisión se espera una recepción.
  - Por último la rutina de atención a la interrupción del módulo I2c gestiona todos los eventos de recepción y transmisión de la línea de comunicación con el micro maestro. Esta función es la responsable de la gestión y construcción de los mensajes con el protocolo i2c correcto.

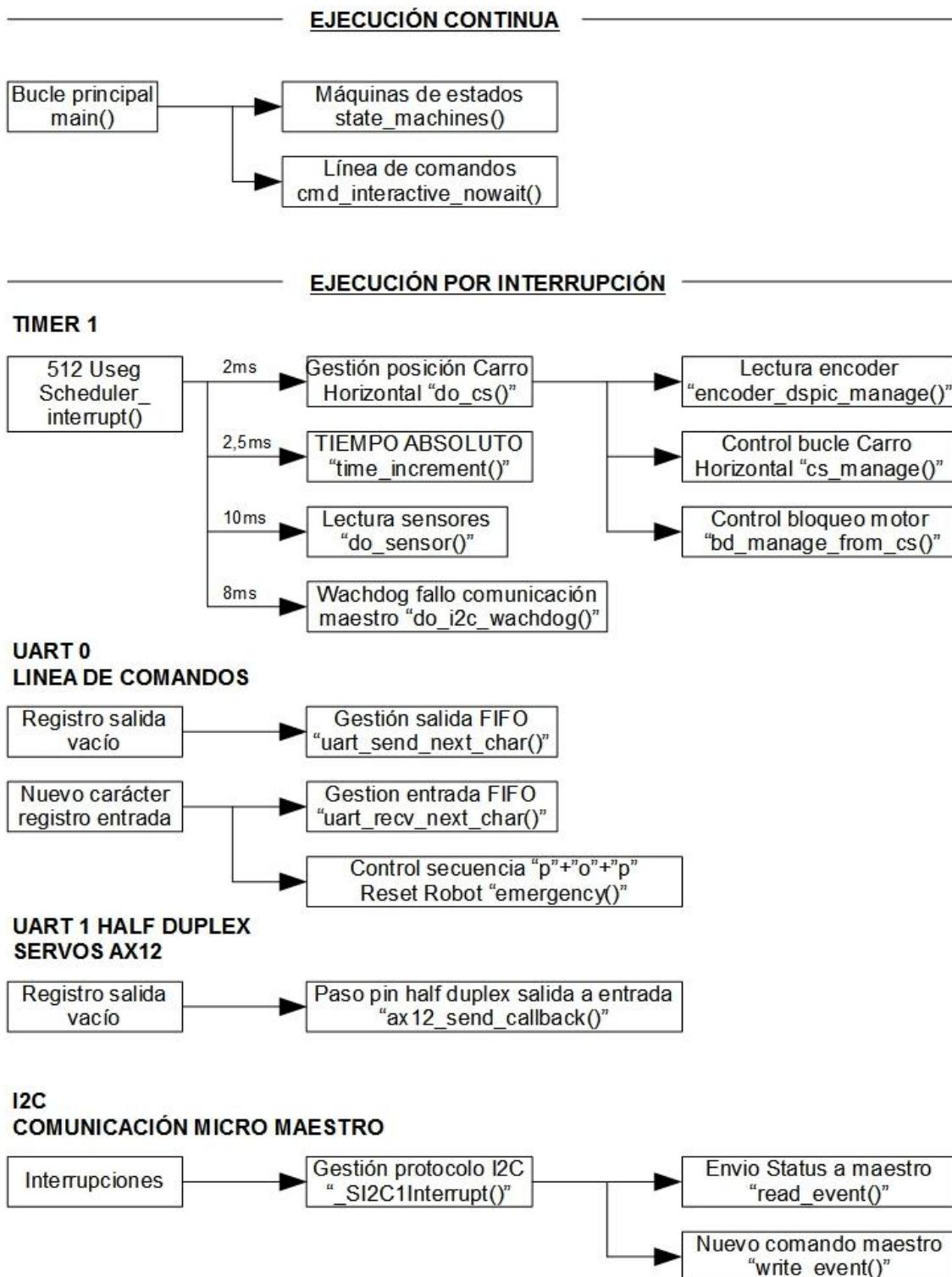


Diagrama 3-7 Diagrama de flujo de tareas del proyecto

### **3.7. Gestión de scheduler. Tareas de ejecución periódica**

En los dsPIC el timer encargado de dar soporte al scheduler es el Timer 1. La interrupción periódica de este Timer es la encargada de ejecutar la función “scheduler\_interrupt(void)” que gestiona el control de los eventos.

Se ha pre-configurado el módulo para que la configuración del Timer1 no se realice de forma automática desde la función “scheduler\_init()” de inicialización de este. Para ello no se ha definido la macro “CONFIG\_MODULE\_SCHEDULER\_TIMER1”.

Para la configuración del Timer 1 se ha creado una función específica en el archivo “main.c” denominada timer\_init(). Esta función se limita a la configuración exclusiva de este timer y no recibe ni devuelve ningún parámetro. Posteriormente es invocada en la zona inicial de configuración en main().

Dentro de esta función se configura el timer para que provoque una interrupción periódica, ajustando el registro de comparación “PR1” con el número de incrementos (cuentas) del timer por periodo de ejecución de la función de interrupción del Scheduler. Se ha implementado la siguiente fórmula para el ajuste automático, en función de la variable FCY y del valor de la macro “SCHEDULER\_UNIT” que ha sido previamente ajustada en el archivo de configuración del módulo “scheduler\_config.h” con el valor de los microsegundos del periodo de interrupción. Este valor se ha ajustado a 512 microsegundos.

$$\text{PR1}(\text{cuentas\_Timer}/\text{T\_Sched}) = \text{SCHEDULER\_UNIT}(\text{useg}/\text{T\_Sched}) * (\text{FCY}(\text{cuentas\_Timer}/\text{seg})/1000000(\text{useg}/\text{seg}))$$

Con:

Prescaler del Timer ajustado a 1:1

SCHEDULER\_UNIT = 512 (microsegundos / Periodo Scheduler)

FCY = FOSC / 2 = 20 Mhz

PR1 = 10240 (cuentas\_Timer/T\_Sched)

Por último esta función de inicialización habilita el Timer y por lo tanto las tareas programadas en el Scheduler.

También se ha creado la función de atención a la interrupción asociada al Timer 1. En esta función se invoca exclusivamente la función de gestión de eventos del Scheduler “scheduler\_interrupt()”.

La programación de las tareas que realizara de forma periódica el timer, se ha realizado en la zona de configuración de main con la función “scheduler\_add\_periodical\_event\_priority” que ya ha sido explicada en las Aversive. Las tareas programadas han sido detalladas en Tabla 3-1.

Función	Periodo	Prioridad
do_cs	2ms	100
do_i2c_watchdog	8ms	110
do_sensors	10ms	120

Tabla 3-1 *Tareas programadas en Scheduler con periodo y nivel de prioridad de ejecución*

Como se puede ver en el Diagrama 3-7 de líneas de ejecución se ejecuta una cuarta tarea de gestión de la variable de tiempo absoluta del sistema. Esta función es programada automáticamente en la función de inicialización de este módulo que es invocada también en la zona de inicialización de main.

## 3.8. Configuración módulos Aversive

Asociados a algunos módulos de la librería Aversive existen unos ficheros de configuración de dichos módulos. En estos ficheros se encuentran agrupadas aquellas variables que deben ser ajustadas por el usuario de la librería para adaptarlos a su sistema. A continuación se detallan solo los archivos que han tenido que ser reconfigurados.

### 3.8.1. Scheduler\_config.h

En este archivo han tenido que ser ajustado el periodo de ejecución del scheduler

SCHEDULER\_UNIT\_FLOAT 512.0

SCHEDULER\_UNIT 512 (microsegundos)

### 3.8.2. Time\_config.h

En este archivo ha tenido que ser ajustada la precisión del contador de tiempo absoluto del sistema.

TIME\_PRECISION 2560

Este valor se ha calculado en base al valor de SCHEDULER\_UNIT, siendo 5 veces el valor de este.

Como se explico para este módulo de las Aversive, es imprescindible que sea múltiplo de la base de tiempo del Scheduler ya que para calcular el tiempo transcurrido en cada ejecución de la función “time” es necesario dividir la precisión del contador entre la base de tiempos de Scheduler y hacer una conversión cast a tipo “int” del resultado, ocurriendo una pérdida de precisión si ambos no son múltiplos.

### **3.8.3. Uart\_config.h**

---

En este archivo los nombres asignados a las macros de las dos Uart que pueden ser configuradas conservan el nombre original de las Aversive, en su adaptación a los dsPIC la Uart 0 corresponde a la 1 del dsPIC y la Uart 1 a la 2 del dsPIC.

- Configuración de Uart 1 de dsPIC para comunicación con PC a través del micro maestro.
- Habilitación de compilación de código “UART0\_COMPILE”.
- Habilitación de interrupción “UART0\_INTERRUPT\_ENABLED”
- 115200 bps “UART0\_BAUDRATE 115200”
- Tamaño de buffer fifo de salida de 64 byte “UART0\_RX\_FIFO\_SIZE”
- Tamaño de buffer fifo de entrada de 14 byte “UART0\_TX\_FIFO\_SIZE”
- 8 bit de datos “UART0\_NBITS”
- Sin paridad “UART0\_PARITY”
- 1 bit de stop “UART0\_STOP\_BIT”
- Configuración de Uart 2 de dsPIC para comunicación (simulada half dúplex) con servos AX12.
- Habilitación de compilación de código “UART1\_COMPILE”.
- Habilitación de interrupción “UART1\_INTERRUPT\_ENABLED”
- 1000000 bps “UART1\_BAUDRATE 1000000”
- Tamaño de buffer fifo de salida de 127 byte “UART1\_RX\_FIFO\_SIZE”
- Tamaño de buffer fifo de entrada de 127 byte “UART1\_TX\_FIFO\_SIZE”
- 8 bit de datos “UART1\_NBITS”
- Sin paridad “UART1\_PARITY”
- 1 bit de stop “UART1\_STOP\_BIT”

### **3.8.4. Pwm\_servo\_config.h**

---

En este archivo se configura la pwm que se aplicara a los servos analógicos.

Frecuencia de señal generada 50 Hz “PWM\_FREQ 50”.

Timer 2 asociado a salidas de comparación “TIMER\_SOURCE 2”.

## **3.9. Codificación comandos slavedspic uart PC**

---

En el apartado 5.4.2. “Comandos uart PC de depuración y configuración” dentro de la sección de especificaciones software se encuentra la lista completa de los comandos que tienen que ser añadidos a los interpretados desde la línea de comandos de la librería IHM. La codificación de cada nuevo comando sigue una misma estructura que va a ser detallada a continuación de forma generalizada.

Todos los comandos codificados se encuentran agrupados en el archivo “commands\_slavedspic.c”.

Los pasos para su configuración son los siguientes:

- Codificación de nuevos comandos para su interpretación, en los archivos de definición de comandos.  
Todo lo necesario para la definición de un comando se encuentra agrupado en el código con la siguiente estructura:
  - Declaración de tipo de variable estructura con datos para comando codificado. La estructura es rellenada cuando el comando es recibido correctamente, con los datos del comando que serán pasados como parámetros a la función del comando.
  - Función de comando recibido. Esta función es ejecutada cuando el comando es recibido con el formato correcto y recibe como parámetros la estructura del comando con los datos recibidos, opcionalmente está preparada para recibir un segundo parámetro. En esta función se analizan los parámetros definidos y se ejecutan las acciones correspondientes según estos.
  - Definición de la cadena de caracteres que será recibida como primer elemento de la línea, corresponde al nombre del comando.
  - Definición de una cadena de caracteres que integre con el siguiente formato todas las cadenas de caracteres que pueden ser introducidas como segundo elemento. “opción\_1”#“opción\_2”#“opción\_3”#....“opción\_n”
  - Opcionalmente se pueden seguir definiendo tantos elementos como vaya a recibir el comando.
  - Cadena de caracteres con texto para ser impreso con comando de ayuda.
  - Declaración e inicialización de estructura de codificación de comando para las funciones de análisis gramatical. Esta incluye:
    - Puntero a función de ejecución de comando recibido.
    - Puntero a segundo argumento que será pasado a la función de ejecución de comando recibido.
    - Puntero a cadena de caracteres de texto de comando de ayuda.
    - Array de punteros a estructuras de fichas de elementos de línea. Corresponde a un array que es generado en los puntos 1.c, 1.d y 1.e.
  - Por último declaración como extern de la estructura del punto 1.g en el archivo “comands.c” e inclusión en array de estructuras de este tipo, para hacerla accesible desde el fichero “cmdline.c”.
- Definición de número de uart como entrada salida de linea de comandos, configuración de la uart en “uart\_config.h” e inicialización de uarts. Por orden serían:
  - #define CMDLINE\_UART “número de uart”. En el dsPIC la uart 1 esta codificado como 0 y la uart 2 como 1.
  - #define UART“número de uart”\_COMPILE, etc.
  - uart\_init();
- Inicialización de sistema con cmdline\_init() y llamada a cmdline\_interact\_nowait() de forma periódica.

### **3.10. Gestión de eventos I2C de comandos maestro**

---

Como se ha explicado en el apartado 1.5 "comm/i2c\_slave" la función de la interrupción del módulo I2c es el responsable de la gestión y construcción de los mensajes con el protocolo i2c correcto. El usuario de la librería solo tiene que definir las funciones de las operaciones de lectura o escritura que serán invocadas desde esta cada vez que sea recibido un byte de datos (no de dirección) con las siguientes particularidades:

- La función de la operación de solicitud de lectura de registro será invocada cada vez que se reciba el primer byte de datos en una operación i2c de escritura. Para comprender regresar al apartado 1.5. Esta función recibe como parámetros ya definidos el byte recibido que corresponde al tipo de operación, siendo las dos posibles “I2C\_REQ\_SLAVE\_DSPI\_C\_STATUS = 1” para operaciones de lectura y “I2C\_CMD\_GENERIC = 0” para operaciones de escritura. Además recibe como parámetro el array en que se deberá guardar la estructura de estado del sistema, si el primer byte recibido y pasado como parámetro corresponde con I2C\_REQ\_SLAVE\_DSPI\_C\_STATUS. La función de atención de interrupción de I2c será la encargada de gestionar el envío de esta estructura byte a byte en posteriores solicitudes de lectura.
- La función de la operación de solicitud de escritura de registro será invocada cada vez que sea recibido un byte de datos en una operación i2c de escritura. Recibirá como parámetros el primer byte recibido que corresponde a lo ya explicado, un array con los datos recibidos que corresponderá a la dirección del segundo byte recibido en el array de recepción y el número de datos que han sido introducidos desde esta dirección.

Estas dos funciones han sido creadas en el nuevo archivo “i2c\_protocol.c” y van a ser detalladas en los dos siguientes subapartados.

En el archivo “i2c\_commands.h” han sido definidas las dos estructuras de datos que pueden ser enviadas y recibidas y que ya han sido explicadas en detalle en el apartado 1 “Protocolo de comunicación I2C entre microcontroladores”, ya que son una especificación de diseño.

Toda la configuración necesaria para esta librería se encuentra en el archivo main en la zona inicial de configuración de la función main:

- Inicialización de i2c con la asignación de la dirección del esclavo en el bus “i2c\_init(I2C\_SLAVE\_DSPI\_C\_ADDR)” con I2C\_SLAVE\_DSPI\_C\_ADDR = 10.
- Asignación de la función que se ejecutara en operaciones de solicitud de lectura “i2c\_register\_read\_event(i2c\_read\_event)”.
- Asignación de la función que se ejecutara en operaciones de solicitud de escritura “i2c\_register\_write\_event(i2c\_write\_event)”.

### **3.10.1. i2c\_write\_event( )**

Los parámetros que recibe esta función han sido definidos previamente en el prototipo de la función `i2c_register_write_event`, y son por lo tanto una especificación para su construcción.

Son por orden:

- “`uint8_t cmd_byte`”
- “`uint8_t *buf`”
- “`int16_t size`”

Ya han sido comentadas de forma individual en el apartado anterior.

Las tareas realizadas en esta función son muy sencillas y para su comprensión no es necesaria la creación de un diagrama de flujo.

Por orden su secuencia de ejecución es la siguiente:

1. Se comprueba que en el primer parámetro se ha recibido “`I2C_CMD_GENERIC`”, que corresponde al código para operaciones de escritura de un nuevo comando desde el micro maestro. En caso contrario se sale de la función.

2. A continuación en función de la clase de estructura de datos recibida, que se encuentra codificada en la primera posición de la estructura de datos recibida (a partir del segundo byte de datos en escritura i2c), se realiza la acción correspondiente con el array pasado como parámetro.

Aunque según las especificaciones software tan solo se va a enviar una sola clase de estructura de datos, se ha creado este sistema para permitir el envío de otras estructuras según necesidades futuras.

El código asignado a la clase de estructura de datos del tipo comando corresponde a “`I2C_CMD_SLAVEDSPIC_SET_MODE = 2`”

3. Cuando se recibe la estructura de comando se comprueba si el tamaño de los datos recibidos hasta el momento corresponden a los de dicha estructura. En caso negativo se sale de la función a la espera de que se complete la recepción, y en caso positivo se almacenan los datos recibidos ya en una estructura con formato de comando y se la pasa como parámetro a la función de cambio de comando.

Esta función de cambio de comando cargará esta estructura en la variable de comando del sistema “`mainboard_command`”, a la que acceden las funciones de estado del sistema para ejecutar las acciones solicitadas.

### **3.10.2. i2c\_read\_event( )**

---

Los parámetros que recibe esta función han sido definidos previamente en el prototipo de la función `i2c_register_read_event`, y son por lo tanto una especificación para su construcción.

Son por orden:

- “`uint8_t cmd_byte`”
- “`uint8_t *buf`”

Ya han sido comentadas de forma individual en el apartado 5.9.

Las tareas realizadas en esta función son muy sencillas y para su comprensión no es necesaria la creación de un diagrama de flujo.

Por orden su secuencia de ejecución es la siguiente:

1. Se comprueba que en el primer parámetro se ha recibido “`I2C_REQ_SLAVEDESPIC_STATUS`”, que corresponde al código para operaciones de lectura de un nuevo comando desde el micro maestro. En caso contrario se sale de la función.
2. Se crea un puntero a una estructura de datos de estado tipo “`i2c_slavedspic_status`” y se carga con la dirección del array de transmisión que a sido pasado como parámetro. Utilizando este puntero se cargan las variables de estado del sistema con el formato de la estructura de estado en el array de transmisión del módulo I2c.
3. Se ajusta a 5ms nuevamente el contador del sistema de watchdog de última lectura de estado del maestro, para que comience la temporización de seguridad nuevamente.

## **3.11. Funciones de gestión de nuevo comando**

---

Cada vez que se recibe correctamente un comando desde el micro maestro a través del módulo I2c o desde el PC a través de la Uart 1, se llama a estas funciones para que gestionen la activación del nuevo comando solicitado. En el Diagrama 3-8 se ha esquematizado la llamada de estas funciones desde los dos módulos de comunicación que hacen uso de ellas.

Como se puede ver en el diagrama de flujo las dos fuentes de entrada de comandos hacen uso de la misma función para copiar la estructura del comando recibida en la estructura de comando del sistema.

La forma de recibir y gestionar el comando es diferente según la fuente de entrada de que se trate. En el módulo i2c los datos almacenados en el array de entrada son pasados hasta la función de cambio de estado direccionándolos con sucesivas conversiones cast de tipo de dato. En las diferentes funciones de atención a los comandos recibidos correctamente por

línea de comandos, se crea una estructura de comando y una vez rellenado con los datos del comando recibido se pasa hasta la función de cambio de estado por referencia.

La función “i2c\_set\_mode” se limita a invocar a la función de cambio de estado.

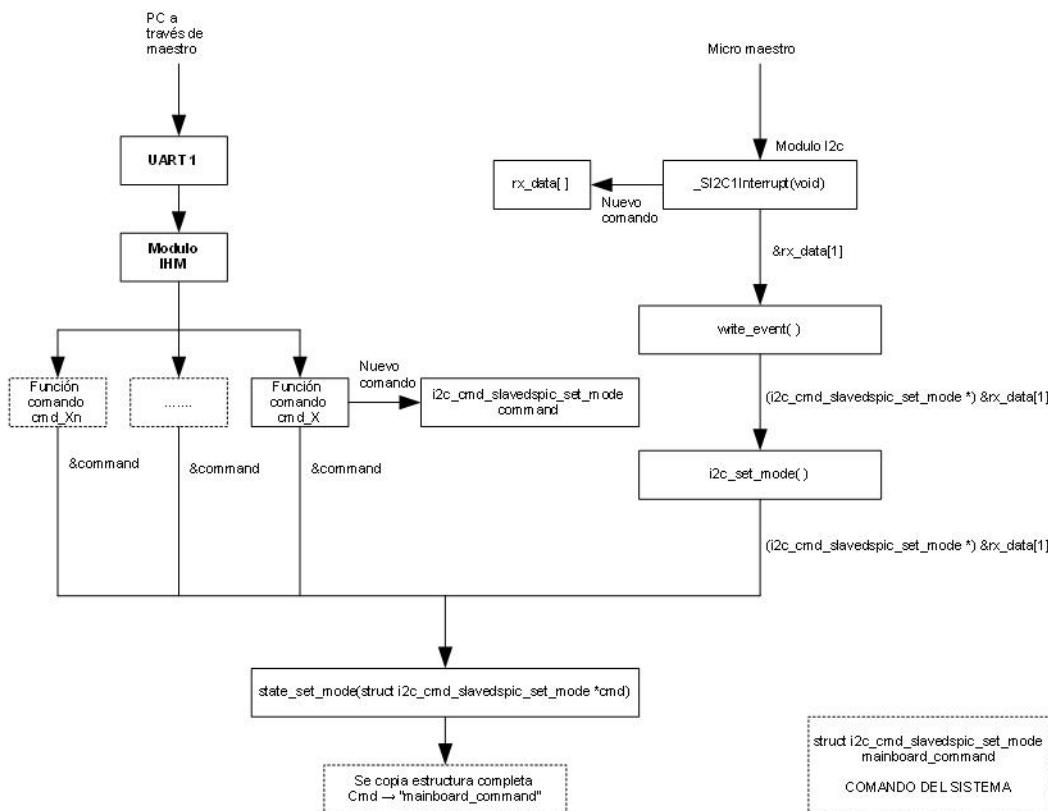


Diagrama 3-8 Diagrama de flujo de orígenes de nuevo comando

### 3.11.1. i2c\_set\_mode( )

En esta función básicamente se copia el comando recibido del array de recepción o de la estructura temporal en la estructura de comando del sistema. También se indica que ha sido cargado un nuevo comando para su ejecución mediante la activación de la variable global “mode\_changed”.

Existen tres excepciones en las que aunque si se carga la estructura de comando no se activa el flag “mode\_changed”. Estas son las siguientes:

- Recepción del comando “POWER\_OFF”

La atención a este comando es inmediata en esta misma función. Se detiene de inmediato cualquier movimiento de servos analógicos, AX12 o motor de carro

horizontal y se cambia la variable de estado del sistema a LISTO para la ejecución de un nuevo comando.

- Recepción de comando para sistema de control de palomitas “I2C\_SLAVE\_DSPI\_C\_MODE\_POPCORN\_SYSTEM”  
Se carga el comando para el sistema de palomitas en la variable de comando solicitado en la estructura de estado del sistema de las palomitas.  
En esta misma estructura se cambia a activo de flag de nuevo comando.
- Recepción de comando para sistema de control de stands “I2C\_SLAVE\_DSPI\_C\_MODE\_STANDS\_SYSTEM”  
Se carga el comando para el sistema de stands en la variable de comando solicitado en la estructura de estado del sistema de stands.  
Se carga el valor del ángulo de la paleta para el sistema de stands en la variable de ángulo de la paleta en la estructura de estado del sistema de stands.  
En esta misma estructura se cambia a activo de flag de nuevo comando.

Tanto el sistema de las palomitas como el de stands no operan con la estructura de comando del sistema, ya que todas las funciones para estos sistemas tan solo operan con los datos recibidos como parámetros, y tan solo reciben la estructura del propio sistema. Es por esto que dentro de su estructura se ha codificado una variable para el comando solicitado y otra de flag para indicar que se ha cambiado el comando y debe ser atendido.

### **3.12. Secuencia de reset por linea de comandos “p o p”**

---

Una de las especificaciones software es la monitorización de los caracteres recibidos por la línea de comandos para el control de la recepción de la secuencia {"p" "o" "p"}, y en cuyo caso se estará indicando al robot que debe resetear el microcontrolador.

Se ha creado una función llamada “emergency( )” que recibe como parámetro el carácter recibido y ha sido programada para su ejecución con cada evento de recepción de byte en la uart 1. Para esto se ha utilizado la función de registro de funciones para evento de recepción de byte “uart\_register\_rx\_event( )”. Esta función ha sido detallada en el apartado 5.1.3. “comm/uart”.

```
uart_register_rx_event(CMDLINE_UART, emergency);
```

La función “emergency” comprueba la recepción de la secuencia mediante una variable declarada como “static”, en la que ha medida que se va sucediendo la secuencia va incrementando su valor. Una vez recibido el primer carácter de la secuencia, con el contador incrementado se esperara la recepción del siguiente carácter de la secuencia. De no sucederse se resetea el contador y se espera la recepción nuevamente del primer carácter.

Una vez incrementado el contador hasta el tercer carácter se resetea el micro.

### **3.13. Función de gestión de registro de errores “mylog( )”**

Para el registro de los errores se ha creado una función única que será ejecutada indistintamente para los 5 niveles de error definidos en el módulo de error de Aversive. Esta función es mylog y ha sido creada en el archivo “cmdline.c” para imprimir los errores registrados a través del puerto serie.

En esta función se ha implementado un sistema para el filtrado de los mensajes a enviar, para facilitar la depuración evitando una impresión excesiva de mensajes.

Se han creado tres niveles de filtrado:

- Por nivel de alarma, mediante la variable “gen.log\_level” que fija el nivel de alarma mínimo para impresión. Con gen.log\_level=4 se imprimen todos los niveles y con gen.log\_level=0 solo se imprime el nivel máximo de alarma. Por defecto el nivel de alarma máximo siempre es impreso.
- Por número de error, solo serán impresos aquellos errores cuyo número de error este almacenado en el array gen.logs[]. Siendo los números de error definidos los siguientes:
  - E\_USER\_I2C\_PROTO 195
  - E\_USER\_SENSOR 196
  - E\_USER\_ST\_MACH 197
  - E\_USER\_CS 198
  - E\_USER\_AX12 199
  - E\_USER\_ACTUATORS 200

Por último esta función también incluye para la impresión a los datos definidos en el array de error el tiempo absoluto en el que es impreso el error.

Para la programación de esta función para los 5 niveles de error se ha incluido en la zona de configuración de la función main, una llamada a cada función de registro de la función asociada a cada nivel, programando en todas esta misma función.

- error\_register\_emerg(mylog);
- error\_register\_error(mylog);
- error\_register\_warning(mylog);
- error\_register\_notice(mylog);
- error\_register\_debug(mylog);

Por último se ha creado una sección de código también en esta zona inicial de configuración donde se habilitan los números de error que serán impresos y el nivel de alarma que filtrara los mensajes imprimibles. El valor de estas variables se ha variado según ha sido necesario durante las labores de depuración. A continuación se muestra un ejemplo.

```
gen.logs[0] = E_USER_ST_MACH; //Se habilita este número de error para la impresión
gen.log_level = 5;           //Se imprimen por la linea de comandos todos los niveles de alarma
```

### **3.14. Configuración y gestión de módulo control de bucle cs para sistema motor-encoder**

El motor del carro de desplazamiento horizontal de stands esta equipado con un encóder con dos canales en cuadratura que ha sido conectado a un módulo de control de encoders del dsPIC. Mediante este encóder se va a realizar un sistema de control de posición en lazo cerrado del carro horizontal. Previo a su utilización será necesario referenciar el sistema chocando contra uno de los límites de desplazamiento mecánico y asignando a esta posición su valor absoluto de referencia. El control del punto de choque se realizará con el sistema de control de bloqueo. Una vez referenciado el sistema se establecerán unos finales de carrera software que limitarán los máximos puntos de desplazamiento programables.

El sistema de control que se ha utilizado está compuesto por los siguientes módulos de Aversive:

- Módulo de control de encóder
- Módulo de control de bucle.
- Módulo de detección de bloqueo.

Cada uno de estos módulos como ya se ha explicado tienen asociada una estructura de datos con todos los parámetros de configuración y variables de estado necesarias para su ejecución. Se ha declarado una estructura llamada “stands\_exchanger” del tipo “cs\_block” que contiene todas las estructuras de datos de los sistemas necesarios, incluidos los filtros que van a ser utilizados en el control de bucle y tres variables de estado. Se detalla su uso a continuación:

- uint8\_t on. Flag que a 1 habilita la ejecución del sistema de control de bucle.
- uint8\_t calibrated. Flag que a 1 indica que el sistema ha sido referenciado según se ha explicado.
- uint8\_t blocking. Flag que es activado a 1 cuando el motor se ha detenido por un bloqueo. Es activado manualmente cuando en la función “stands\_exchanger\_check\_position\_reached( )” para la comprobación de posición alcanzada es detectado el bloqueo del motor. Es desactivada cada vez que se programa una nueva posición de desplazamiento con la función stands\_exchanger\_set\_position( ).
- struct cs cs. Ver apartado 5.1.6. con la descripción detallada.
- struct pid\_filter pid.
- struct quadramp\_filter qr.
- struct blocking\_detection bd.

### **3.15. Función de gestión de módulos “do\_cs( )”**

Los tres módulos principales que conforman el sistema, necesitan la ejecución periódica de sus funciones de control. Para gestionar esta ejecución de forma simultánea se ha creado la función “do\_cs( )” que invoca a estas tres funciones con la misma frecuencia, cada 5 milisegundos. La ejecución de esta función se ha programado como una tarea periódica en el

módulo Scheduler como ha sido explicado en el apartado 3.7 "Gestión de scheduler. Tareas de ejecución periódica".

Es en esta función donde se inhibe la ejecución de la función de control de bucle con la desactivación del flag “stands\_exchanger.on” que ha sido explicado en el apartado anterior.

### 3.16. Estructura de control de bucle cs

En el Diagrama 3-9 se muestran los bloques que han sido configurados para su funcionamiento dentro del sistema de control de bucles de Aversive. Dentro de cada uno de los bloques ya se adelanta la función de trasferencia resultante con la configuración finalmente ajustada. También se indica el tipo de dato que recibe cada bloque y su unidad.

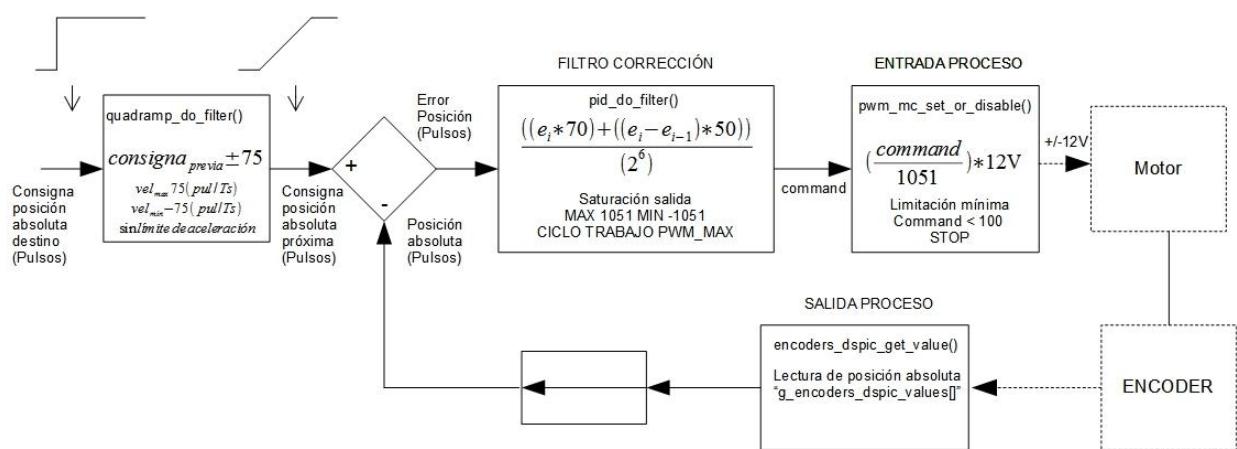


Diagrama 3-9 *Diagrama de esquematización de la configuración del módulo de control de bucle para el control del motor del carro horizontal*

Los bloques generales de un sistema de control en Aversive son los siguientes:

- **FILTRO DE CONSIGNA.**  
Se ha utilizado un filtro Quadramp, que es un filtro específico para filtros de consigna. Este filtro ha permitido limitar la variación máxima de consigna a su salida, estableciendo un sistema de control de velocidad y aceleración del movimiento. Como se explico carece de realimentación y por lo tanto no asegura la velocidad conseguida y puede crear reacciones violentas en caso de bloqueo, por la acumulación de la diferencia de error.
- **BLOQUE DE SALIDA DE PROCEDO**  
Este bloque que es creado por el usuario de la librería, va a consistir en una función “encóder\_dspic\_get\_value( )” que accede a la variable de posición absoluta del módulo de control del encóder y la devuelve al sistema de control de bucle.

- **FILTRO DE CORRECCIÓN**

Se ha utilizado un filtro PID. Este filtro va a recibir el error de distancia en pulsos de encóder y devuelve el comando de movimiento saturado a los valores máximos de ajuste del ciclo de trabajo para la generación de la PWM del módulo de control de motores.

- **BLOQUE DE ENTRADA EN PROCESO**

Este bloque que es también creado por el usuario de la librería, va a consistir en una función “pwm\_mc\_set\_or\_disable( )” que recibirá el valor devuelto por el filtro de corrección y un puntero a la estructura de datos del módulo de control de motores por PWM, donde se encuentran todos sus datos de configuración.

Dentro de esta función se ha implementado un sistema para bloquear el movimiento con consignas por debajo de un umbral. Se ha definido este umbral con la macro “PWM\_DISABLE\_VALUE\_TH” y ha sido fijado al valor “100”. Utilizando la función de transferencia de este bloque del diagrama 14, se calcula que no se generarán consignas de movimiento con alimentaciones equivalentes en el motor, inferiores en valor absoluto a 1,14 V.

$$V_{maxmin} = (\pm 100/1051)12V$$

Este sistema va a eliminar los pequeños movimientos de ajuste, estabilizando y haciendo más sencilla la configuración del sistema.

Desde esta función finalmente se invoca a la función “pwm\_mc\_set( )” donde se ajusta el registro de comparación para la generación de la pwm del módulo de control de motores.

- **FILTRO DE REALIMENTACIÓN.**

Este filtro corresponde a la caja que en el Diagrama 3-9 se ha representado vacía. No se va a utilizar ningún filtro de aplicación en esta posición del bucle.

### **3.16.1. Función de configuración de módulo cs “slavedspic\_cs\_init( )”**

Se ha creado la función “slavedspic\_cs\_init( )” para agrupar en ella toda la configuración que es necesario realizar sobre los distintos elementos del sistema de control de bucle. A continuación se detalla la configuración aplicada y en los apartados próximos se van a desarrollar los procesos de selección y ajuste empleados.

- **Configuraciones filtro de corrección PID**

- Mediante la función “pid\_init( )” se ponen a cero todos los byte de la estructura declarada para el filtro. Se inicializa a 1 la constante de ganancia proporcional y se ajusta a 1 iteración el periodo de cálculo del error derivativo.
- Con la función “pid\_set\_gains( )” se ajustan en la estructura del filtro las constantes PID a Kp=70, Ki=0 y Kd=50.
- Con “pid\_set\_maximums( )” se fijan los límites de saturación a INmax=0 (valor máximo de entrada a filtro no limitado), Imax=1051 (valor de error

acumulativo integral), OUTmax=1051 (valor absoluto máximo de salida del módulo).

- Con “pid\_set\_out\_shift()” se ajusta el valor de la constante de adaptación de escala a OUTshift=6. Este número corresponde con el número de veces que se desplazan a la derecha los bits de la variable de salida del módulo. Por lo tanto equivale a dividir su valor por 2 elevado a OUTshift, y con el ajuste realizado a dividir por 64.
- Con “pid\_set\_derivate\_filter( )” se ha ajustado a una iteración el periodo empleado para el cálculo del error derivativo. Este se calculará restando al error de entrada en la iteración anterior, el error de la iteración.

- **Configuraciones filtro de consigna Quadramp**

- Mediante la función “quadramp\_init( )” se ponen a cero todos los byte de la estructura declarada para el filtro y que es pasada por referencia como parámetro.
- Con la función “quadramp\_set\_1st\_order\_vars( )” se ajustan en la estructura del filtro las constantes de primer orden positiva “var\_1st\_ord\_pos=75” y negativa “var\_1st\_ord\_neg=75”. Este valor corresponde con la máxima variación de la consigna que será pasada a la salida del filtro e introducida en el sistema.
- Con la función “quadramp\_set\_2st\_order\_vars( )” se ajustan en la estructura del filtro las constantes de segundo orden positiva “var\_2st\_ord\_pos=0” y negativa “var\_2st\_ord\_neg=0”. Este valor corresponde al máximo incremento de la variación de la consigna a la salida (aceleración). A 0 significa que no ha sido limitado, siempre se aplicará la velocidad máxima.

- **Configuración sistema de control de bucle**

- Mediante la función “cs\_init( )” se borra toda la información almacenada en la estructura de datos de control de bucle. En esta estructura como ya se ha explicado en el apartado 5.1.6. tenemos punteros a las funciones que se ejecutarán por cada uno de los bloques del sistema y a las estructuras de datos de cada uno de los bloques, además de las variables de estado y funcionamiento. Los punteros pasan a apuntar a NULL y las variables contendrán 0.
- Mediante las funciones de registro de cada uno de los bloques del sistema se almacena en la estructura del sistema de control de bucle la dirección de la función que se ejecutara en cada posición del bucle y la estructura de datos donde se almacena la configuración de cada uno de ellos. A continuación se muestra el código introducido, donde el primer parámetro corresponde a la estructura del sistema de control de bucle, el segundo a la dirección de la función del filtro o módulo que se asigna y la tercera la dirección de la estructura de datos asociada al bloque de la función asignada.

```
cs_set_consign_filter(&slavedspic.stands_exchanger.cs.quadramp_do_filter,  
&slavedspic.stands_exchanger.qr);  
cs_set_correct_filter(&slavedspic.stands_exchanger.cs.pid_do_filter,  
&slavedspic.stands_exchanger.pid);  
cs_set_process_in(&slavedspic.stands_exchanger.cs.pwm_mc_set_or_disable,  
PWM_MC_STANDS_EXCHANGER_MOTOR);  
cs_set_process_out(&slavedspic.stands_exchanger.cs.encoders_dspic_get_value,  
STANDS_EXCHANGER_ENCODER);
```

- Por último mediante la función “cs\_set\_consign( )” de ajuste de valor de consigna del sistema se inicia el valor de esta a 0, que es también el valor que se asigna en el sistema de control del encóder a la posición inicial en la inicialización.

- **Configuración sistema de control de bloqueo de motor**

- Mediante la función “cs\_init( )” se resetea la estructura de datos apuntada por “blocking\_detection”. Pone todos los bits a cero.
- Con la función “bd\_set\_speed\_threshold( )” se ajusta a 50 pul/iteración la velocidad por encima de la cual no se detectara bloqueo, quedando el detector deshabilitado.
- Con “bd\_set\_current\_thresholds( )” se ajustan por orden:
  - Constante K1 = 800. Contante de la formula de estimación de la corriente en función de la tensión de excitación de este y las rpm de giro del eje del motor.
  - Constante k2 = 10,000. Contante de la formula de estimación de la corriente en función de la tensión de excitación de este y las rpm de giro del eje del motor.
  - Umbral de corriente = 1,000,000 (micro-amperios). Se fija en un amperio la corriente máxima de trabajo, por encima de la cual se considera que el motor se encuentra bloqueado.
  - Iteraciones de umbral de corriente = 10. Número de iteraciones que tendrá que superarse de forma consecutiva el umbral de corriente para considerarse bloqueo.
- Por último se termina habilitando la ejecución del sistema de control de bucle activando “stands\_exchanger.on = 1”.

Esta función es invocada en la zona inicial de configuración de la función main.

Y para completar la configuración del sistema también son invocadas las funciones de configuración de los módulos de entrada y salida al sistema:

- **Módulo de ENCODER**

- Mediante la función “encoders\_dspic\_init()” que no recibe parámetros se configuran los registros de las entradas de encóder que se encuentren

habilitadas por la declaración de la macro correspondiente. Para esta aplicación ha sido habilitada la entrada de encóder número 1 con la definición de “ENCODERS\_DSPIC1\_ENABLED”. Los pines fueron configurados y remapeados en la función de configuración de pines del micro.

- Finalmente esta función hace una primera lectura del encóder y inicializa el valor de la variable de posición absoluta del sistema, “g\_encoders\_dspic\_values[0]” para la entrada número 1 de encóder. Esta variable es de uso global en el archivo “encóder\_ds\_pic.c”. Para acceder a ella desde el exterior se utilizan las funciones específicas con zona protegida libre de interrupciones.

- **Módulo de PWM para motores**

- Mediante la función “pwm\_mc\_channel\_init( )” se carga en la estructura de datos del módulo pwm de control de motores la siguiente configuración del módulo:

Modo de funcionamiento bipolar. En este modo la tensión en el motor conmuta de +12 a -12V que es la tensión de alimentación que se ha aplicado al puente en H, en el transcurso del periodo según sea el ciclo de trabajo ajustado. Con un ciclo del 50% la tensión media resultante que ve el motor es de 0V.

Canal número 1.

Módulo pwm número 2.

- Mediante la función “pwm\_mc\_init( )” se realiza toda la configuración del módulo codificada en la estructura que se pasa como primer parámetro y se establece la frecuencia de trabajo en 19KHz. También se establecen los márgenes máximos y mínimos de la consigna de PWM que puede recibir el bloque de entrada de proceso del sistema de control de bucle, calculada en función del modo de funcionamiento del generador de la pwm. También se calcula el offset que se aplicara para los modos bipolar y con signo. Por último esta función configura los pines asociados a este, como salida y entrada del canal en uso.

Para la elección de la frecuencia de trabajo elegida se han tenido en cuenta los siguientes criterios:

1. El periodo de la frecuencia de control es al menos 10 veces menor que la constante de tiempo del motor. Esto garantiza un correcto filtrado de la tensión media.

*Con:*

$$T_{pwm} = (1/19\text{Khz}) = 52,63\mu\text{s}$$

$$R_m = 0.492\text{mH}$$

$$L_m = 5.61\Omega$$

$$\tau_e = L_m/R_m = 0.492 * 10^{-3}/5.61 = 87.7\mu\text{s}$$

$$\tau_e/10 = 8.77\mu\text{s} > T_{pwm}$$

2. El Tpwm tiene que ser mayor que el tiempo de respuesta de los dispositivos de conmutación.

Como se ve en la Tabla 3-2 el tiempo de respuesta acumulado de todos los dispositivos en el peor caso es muy inferior al periodo de la pwm.

Dispositivo	Referencia	Parámetro	T máximo	Trise TOTAL	Tfall TOTAL		
Driver mosfet	TC4427	Trise	30ns	30			
		Tfall	30ns		30		
		Delay time	50ns	50	50		
Mosfet HEXFET	IRF7389	Td(on) Delay	N-Ch-12ns	12			
			P-Ch-19ns	19			
		Trise	N-Ch-13ns	13			
			P-Ch-20ns	20			
		Td(off) Delay	N-Ch-39ns		39		
			P-Ch-51ns		51		
		Tfall	N-Ch-26ns		26		
			P-Ch-48ns		48		
			Canal N	105ns	145ns		
			Canal P	119ns	179ns		

Tabla 3-2 *Tiempos de respuesta máximos del driver TC4427 mas el mosfet IRF7389* (Microchip Technology Inc., 2006) (International Rectifier, 2004)

3. El periodo de la PWM es menor de 50 microsegundos con lo que se evita la aparición del ruido audible, que puede ser muy molesto.

Mediante la función “pwm\_mc\_set( )” se ajusta la tensión de salida del módulo a 0 voltios.

### 3.16.1.1.Cálculo de coeficiente de velocidad y aceleración

Mediante estos coeficientes se relaciona la velocidad y aceleración entre las unidades de sistema y las unidades de usuario, que voy a definirlas en el sistema internacional.

El sistema de movimiento del carro horizontal esta compuesto por los siguientes elementos:

- Encoder de 100 pul/rev y dos canales en cuadratura con conexión directa al eje de motor.
- Motor DC conectado a reductora de engranajes.
- Reductora de engranajes con reducción 14:1 conectada a polea dentada con diámetro primitivo = 9,6mm.
- Correa dentada entre dos poleas con conexión de arrastre a carro horizontal.

Calculo de milímetros de avance de carro por revolución de polea:

$$mm/rev = D_p \pi = 9,6\pi = 30,16 mm/rev$$

Calculo pulsos de encóder por revolución de polea, teniendo en cuenta que los pulsos del encóder se multiplican por cuatro en el módulo con dos canales en cuadratura.

$$pul/rev = 400(pul/rev_{enc}) * 14(rev_{enc}/rev) = 5600 pul/rev$$

Relacionando las dos expresiones, se calculan los pulsos por milímetro de avance del carro horizontal.

$$\frac{5600(pul/rev)}{30,16(mm/rev)} = 185,67 pul/mm$$

Multiplicando esta expresión por los segundos por periodo de muestreo “la unidad”.

$$185,67 pul/mm * 0,002 s/T_i * 1000 mm/m = 371,34 \frac{(pul/T_i)}{(m/s)}$$

$$Coef. Velocidad = 371,35 \frac{(pul/T_i)}{(m/s)}$$

Calculo coeficiente aceleración:

$$371,35 \frac{(pul/T_i)}{(m/s)} * 0,002 s/T_i = 0,74268 \frac{(pul/T_i^2)}{(m/s^2)}$$

### 3.16.1.2. Configuración filtro de consigna “Quadramp”

El carro horizontal tiene un margen de desplazamiento total de 278 mm de recorrido entre topes mecánicos. En su función de construcción del foco realiza la mitad del desplazamiento y regresa para arrastrar un nuevo stand.

Según las especificaciones mecánicas el tiempo de este desplazamiento no debe de superar 0,75 segundos.

Calculo de velocidad mínima requerida.

$$v = \frac{0,278/2 m}{0,75 s} = 0,1853 m/s$$

Conversión a sistema de unidades de robot.

$$v = 0,1853(m/s) * 371,35 \frac{(pul/T_i)}{(m/s)} = 68,81(pul/T_i)$$

Después de realizar diferentes pruebas, fijando esta velocidad máxima en el filtro y sin aplicar límite de aceleración se establece finalmente la velocidad en 75 pulsos por periodo. Se ha calculado el tiempo teórico de desplazamiento para esta nueva velocidad.

$$v = \frac{75(pul/T_i)}{371,35 \frac{(pul/T_i)}{(m/s)}} = 0,2019(m/s)$$

$$\frac{0,278/2 (m)}{0,2019(m/s)} = 0,68(s)$$

No se ha fijado límite de aceleración, ya que es interesante el funcionamiento más violento que se pueda conseguir, con el fin de lanzar el stand hacia la otra torre.

### **3.16.1.3. Configuración y ajuste filtro de corrección “PID”**

Debido a las características poco exigentes del sistema mecánico a controlar, las holgadas tolerancias de la especificación de posicionamiento y el reducido tiempo de desarrollo, se ha optado por realizar un ajuste empírico del controlador, apoyado por las funciones de monitorización de las variables del controlador de bucle a través de la línea de comandos. Mediante estas funciones va a ser posible capturar y comparar la respuesta del sistema frente a una señal escalón.

Se ha descartado la utilización de la constante integral, ya que la especificación de posicionamiento tiene un margen de error muy elevado en régimen permanente. Para anular la utilización del error integral se asignado el valor 0 a la constante integral.

Para el ajuste se ha realizado el siguiente proceso de ensayo y error:

1. Ajuste de todas las constantes a cero y excitación del sistema con consigna escalón.
2. Aumento progresivo de la constante proporcional hasta conseguir que el sistema se vuelva muy subamortiguado o incluso oscilante. Este valor correspondió a la ganancia última.
3. Se fija el valor de la constante proporcional a la mitad de la última ganancia.
4. Aumento progresivo desde cero de la constante derivativa, haciendo pruebas sucesivas con cambios de valor de consigna en escalón, hasta conseguir la respuesta deseada.

### **3.16.1.4. Configuración y ajuste de módulo de detección de bloqueo**

Para el ajuste de este sistema se ha aproximado el valor de las constantes mediante el cálculo teórico con los parámetros de motor, y a continuación se han reajustado manualmente hasta conseguir un resultado óptimo. En el apartado de análisis de dicho módulo de las Aversive “5.1.9. blocking\_detection\_manager” se ha analizado la función utilizada para el cálculo de la corriente estimado y se ha aproximado la formula equivalente de estas constantes en condiciones de régimen permanente.

$$i(t) = (1/R)V_i - (K_b/R)w(t)$$

$$\begin{aligned} K1 &= (1/R) \\ K2 &= (K_b/R) \end{aligned}$$

$$i(t) = K1V_i - K2w(t)$$

Datos de motor Maxon DC 2322.980-52.235-200:

$$\begin{aligned} R &= 5,61\Omega \\ L &= 0,492H \\ K_v &= 622 \text{ rpm/V} \end{aligned}$$

Calculo de constante contraelectromotriz:

$$K_b = 1/K_v = 1/622 = 1607 * 10^{-6} V/rpm$$

Análisis de unidades de ecuación de corriente:

$$i(\mu A) = \frac{1}{R(\Omega)} * V_i(\mu V) - \frac{K_b(\mu V/rpm)}{R(\Omega)} * w(rpm)$$

Conversión a unidades del sistema del control de bucle:

$$\begin{aligned} V_i(\mu V)acmd(c) \\ w(rpm)aw(pul/T_i) \end{aligned}$$

*Siendo  $c$  el registro de pwm motor.  
El rango máximo corresponde a 1051 para 12V*

$$\begin{aligned} cmd(c) &= cmd(c) * \frac{12(V)}{1051(c)} * 10^6 (\mu V/V) = cmd(c) * \frac{12 * 10^6}{1051} (\mu V/c) \\ w(pul/T_i) &= w(pul/T_i) * \frac{1(rev)}{400(pul)} * \frac{1(T_i)}{2(ms)} * \frac{60 * 10^3(ms)}{1(min)} = w(pul/T_i) * \frac{30 * 10^3(rev/min)}{400(pul/T_i)} \end{aligned}$$

Uniendo las expresiones:

$$\begin{aligned} i(\mu A) &= \frac{1}{5,61(\Omega)} * cmd(c) * \frac{12 * 10^6}{1051} \left( \frac{\mu V}{c} \right) - \frac{10^6}{622} \left( \frac{\mu V}{rev/min} \right) * \frac{1}{5,61(\Omega)} * w(pul/T_i) \\ &\quad * \frac{30 * 10^3(rev/min)}{400(pul/T_i)} \end{aligned}$$

Expresión de cálculo de la corriente en las unidades del sistema:

$$i(\mu A) = 2035 \left( \frac{\mu V}{\Omega c} \right) * cmd(c) - 21493 \left( \frac{\mu V(T_i/pul)}{\Omega} \right) * w(pul/T_i)$$

$$K_1 = 2035 \left( \frac{\mu V}{\Omega c} \right)$$
$$k_2 = 21493 \left( \frac{\mu V (T_i / \text{pul})}{\Omega} \right)$$

Ajuste de umbral de detección de bloqueo o velocidad máxima sobre la que no se detecta bloqueo para cualquier consumo de corriente. El único inconveniente que se puede plantear es una situación en la que el robot sea capaz de arrastrar a un robot oponente a una velocidad superior a la umbral.

Se ha fijado esta velocidad en 13,5 cm/s después de realizar pruebas de movimiento con el robot, que aplicando el coeficiente de velocidad es aproximadamente 50 (pul/Ts).

La limitación de corriente sobre la que se detectara el bloqueo se ha establecido en 1A. Para su cálculo se han tenido en cuenta las siguientes características del sistema:

*Características motor:*

*Tensión nominal 12V*

*Potencia 6w*

$R_m = 5.61\Omega$

$$I_{maxbloq\_ueo} = 12(V) / 5.61(\Omega) = 2.14(A)$$

$$I_{nom} = 6(w) / 12(V) = 0.5(A)$$

$$I_{umbral} = 2I_{nom} = 2 * 0.5 = 1(A)$$

Finalmente el valor ajustado mediante la función “bd\_set\_speed\_threshold( )” ha sido introducido en las unidades del sistema (uA).

A continuación es necesario fijar el número de iteraciones que deberá superarse el límite de corriente de manera consecutiva para considerarse bloqueo de motor. Mediante este sistema se filtran los posibles picos de consumo en el arranque del motor que pueden superar la corriente umbral.

Se ha estimado este tiempo en 20 ms y se han calculado las iteraciones de umbral de corriente fijando su valor a 10, conociendo el periodo del sistema de control de bloqueo de 2 ms.

## 3.17. Funciones de nivel intermedio para control de actuadores “ACTUATOR.C”

---

### 3.17.1. Funciones de control de servos AX12

---

Asociado a cada servo AX12 del sistema se ha declarado una estructura de datos del tipo “ax12\_traj” en donde se encuentra almacenada toda la información de configuración del servo, y algunas de las variables de proceso. Detallo su significado a continuación:

- “.id” Identificador del servo en la bus I2c.
- “. zero\_offset\_pos” Offset en pulsos a posición central de la carrera de movimiento del servo.
- “. goal\_angle\_deg” Posición de destino de movimiento programada en grados.
- “. goal\_pos” Posición de destino de movimiento programada en pulsos de servo.
- “. goal\_time\_ms” Tiempo calculado para desplazamiento programado con velocidad de desplazamiento programa en servo.
- “. time\_us” Tiempo absoluto del sistema en microsegundos cuando se inicio del movimiento
- “. pos” Variable auxiliar interna para almacenar la posición en pulsos alcanzada cuando la función es invocada.
- “. angle\_deg” Variable auxiliar interna para almacenar la posición en grados alcanzada cuando la función es invocada.
- “. speed” Velocidad de giro programada en el AX12.

- void **ax12\_set\_pos**(struct ax12\_traj \*ax12, int16\_t pos)

Función para programación de movimiento indicando posición destino en pulsos del registro interno del servo. Es necesario conocer el margen max y min de este valor.

*Parámetros:*

- Dirección de estructura del servo sobre la que se aplicara el movimiento.
- Posición destino en pulsos del sistema interno del servo.

**Funcionamiento interno:**

1. Lectura de posición actual del servo y actualización de variables “pos” y “angle\_deg” con la conversión a grados necesaria en la estructura del servo.
2. Actualización de las variables destino “goal\_pos” y “goal\_angle\_deg” con la conversión a grados necesaria en la estructura del servo, con la solicitada.
3. Programación de posición de destino mediante escritura en el registro de destino de servo.
4. Lectura de la velocidad programa en el registro de velocidad del servo y calculo de tiempo de desplazamiento conociendo ángulo actual, ángulo

destino y velocidad. Almacenamiento de dicho valor en la variable “goal\_time\_ms” de la estructura del servo.

5. Lectura de la variable de tiempo global del sistema con desbordamiento y almacenado de esta en la variable “time\_us”, que corresponderá con el tiempo absoluto de inicio del movimiento.

- **void ax12\_set\_a(struct ax12\_traj \*ax12, int16\_t a)**

Función para programación de movimiento indicando posición destino en grados. El valor 0 grados corresponde con la posición central del rango total del desplazamiento del servo.

*Parámetros:*

- Dirección de estructura del servo sobre la que se aplicara el movimiento.
- Posición destino en grados.

*Funcionamiento interno:*

1. Lectura de posición actual del servo y actualización de variables “pos” y “angle\_deg” con la conversión a grados necesaria en la estructura del servo.
2. Actualización de las variables destino “goal\_pos” y “goal\_angle\_deg” con la conversión a grados necesario en la estructura del servo, con la solicitada.
3. Conversión de ángulo de destino de grados a pulsos del sistema del servo.
4. Programación de posición de destino mediante escritura en el registro de destino de servo.
5. Lectura de la velocidad programa en el registro de velocidad del servo y calculo de tiempo de desplazamiento conociendo ángulo actual, ángulo destino y velocidad. Almacenamiento de dicho valor en la variable “goal\_time\_ms” de la estructura del servo.
6. Lectura de la variable de tiempo global del sistema con desbordamiento y almacenado de esta en la variable “time\_us”, que corresponderá con el tiempo absoluto de inicio del movimiento.

- **int16\_t ax12\_get\_a(struct ax12\_traj \*ax12)**

Función que dada una estructura de servo AX12, devuelve la posición actual instantánea en grados.

*Parámetros:*

- Dirección de estructura del servo sobre la que se devolverá la posición.
- *Retorna:* Posición actual del servo en grados.

*Funcionamiento interno:*

1. Lectura de posición actual del servo y actualización de variables “pos” y “angle\_deg” con la conversión a grados necesaria en la estructura del servo.
2. Salida con devolución de posición en grados.

- `uint8_t ax12_test_traj_end(struct ax12_traj *ax12, uint8_t flags)`  
Función que comprueba si el movimiento programado en un servo AX12 ha concluido o si se ha superado el triple del tiempo calculado para el movimiento. Esta función no bloquea la ejecución y devuelve un código de nivel de realización según la cercanía al punto de destino.

*Parámetros:*

- Dirección de estructura del servo sobre el que se devolverá la posición.
- Mediante esta flag se indica que códigos de control de posición serán comprobados. Si se pasa como parámetro la siguiente expresión “END\_NEAR | END\_TRAJ” se habilitara la devolución de ambas cuando sean alcanzadas las posiciones de cada una de ellas.

*Retorna:* Uno de los valores de la Tabla 3-3, donde se codifica la posición y estado actual:

Valor	Macro	Significado
0	“no codificado”	Movimiento no finalizado
1	END_TRAJ	Distancia a destino menor de “AX12_WINDOW_NO_NEAR” (6 grados)
2	END_NEAR	Distancia a destino menor de finalizado “AX12_WINDOW_NEAR” (10 grados)
4	END_TIME	NO UTILIZADA
8	END_BLOCKING	Tiempo de movimiento superó el triple del tiempo programado

Tabla 3-3 Bits de código de estado de conclusión de movimiento de servos

*Funcionamiento interno:*

1. Lectura de posición actual del servo y actualización de variable “pos” en la estructura del servo.
2. Si fue pasado como parámetro el flag “END\_TRAJ” se comprueba si la diferencia al destino es menor de “AX12\_WINDOW\_NO\_NEAR”, y en ese caso se devuelve el código “END\_TRAJ”.
3. Si no se cumplió la anterior condición si fue pasado como parámetro el flag “END\_NEAR” se comprueba si la diferencia al destino es menor de “AX12\_WINDOW\_NEAR”, y en ese caso se devuelve el código “END\_NEAR”.
4. Si no se cumplieron ninguna de las condiciones anteriores si se ha sobrepasado el triple del tiempo calculado para el movimiento, se detiene el movimiento del servo ajustando como destino la posición actual y se devuelve el código “END\_BLOCKING”.
5. Si no se han dado ninguna de las tres condiciones anteriores se devuelve 0 indicando que el movimiento está en curso.

- **uint8\_t ax12\_wait\_traj\_end(struct ax12\_traj \*ax12, uint8\_t flags)**

Función que bloquea la ejecución del programa hasta que la función “ax12\_test\_traj\_end( )” que es invocada desde dentro de esta, devuelve un código distinto de “0”. Esto es, se produzca cualquier condición diferente de “movimiento de servo en ejecución”. Esta función accede a la función “ax12\_test\_traj\_end” cada 5ms. En cada acceso se está haciendo una lectura del registro de posición del servo AX12.

*Parámetros:*

- Dirección de estructura del servo sobre el que se devolverá la posición.
- Mediante este flag se indica que códigos de control de posición serán comprobados. Si se para como parámetro la siguiente expresión “END\_NEAR | END\_TRAJ” se habilitará la devolución de ambas cuando sean alcanzadas las posiciones de cada una de ellas.

*Retorna:* Uno de los valores de la Tabla 5.9, donde se codifica la posición y estado actual.

### **3.17.2. Funciones de control del carro horizontal**

---

Todas las funciones del carro horizontal, trabajan directamente con los datos de la estructura “stands\_exchanger” del tipo “cs\_block”, donde están almacenadas las estructuras de todos los sistemas dependientes. Por lo tanto no ha sido incluido como parámetro.

- **void stands\_exchanger\_hard\_stop(void)**

Función que detiene el movimiento del carro sin rampa de desaceleración. No recibe parámetros ni devuelve nada.

*Funcionamiento interno:*

- Se estable como consigna de entrada al sistema de control de bucle la posición actual del encóder
- Se resetea el filtro Quadramp estableciendo como salida previa del filtro la posición actual del encóder

- **void stands\_exchanger\_calibrate(void)**

Función que referencia la posición del carro horizontal. Este sistema no tiene finales de carrera de límite de carrera mecánica y es necesario mover el carro en una dirección hasta que choque con un límite de recorrido mecánico. Conocida la posición absoluta de este punto en nuestro sistema de coordenadas, una vez producido el choque se le asigna su posición absoluta real. El choque es detectado por el sistema de control de bloqueo de motores.

*Funcionamiento interno:*

1. Se leen las constantes ajustadas en el filtro PID del sistema y son almacenadas en unas variables temporales.
2. Se ajustan las siguientes nuevas constantes para el filtro PID. Kd=50, Ki=0, Kd=0.
3. Se reconfigura el filtro quadramp con los siguientes parámetros, se ha reducido la velocidad para que el choque sea más controlado.

- Límite velocidad positivo y negativa = 20 pul/Ti (53,85mm/s)
  - Límite de aceleración positiva y negativa = 10 pul/Ti<sup>2</sup> (13.46 m/s<sup>2</sup>) Con esta aceleración se alcanzara la velocidad máxima en dos periodos 2\*0.05s.
4. Se inicia el movimiento hasta la posición codificada en “AUTOPOS\_BIG\_DIST\_mm” que ha sido iniciado a “-350” mayor que la carrera total del sistema mecánico, con lo que se terminará produciendo el choque.
  5. Se entra en un bucle de espera de activación de condición de bloqueo de movimiento. Este bucle dispone de un sistema de temporización timeout que limita el tiempo máximo de espera al valor definido por “AUTOPOS\_BD\_TIMEOUT\_ms”, ha sido establecido a 10000 ms. Si se alcanza el time out se informa que la posición de bloqueo no fue alcanzada y se abandona la ejecución, en caso contrario se continua.
  6. Una vez continuada la ejecución se ajusta el valor absoluto del encóder a la posición de bloqueo con el valor definido por “STANDS\_EXCHANGER\_CALIB\_IMP\_MAX”. Cálculo de posición absoluta de bloqueo:

$$\text{Carreratotal} = 278\text{mm} * 185,67 \text{ pul/mm} = 51616,26\text{pul}$$

*El origen(0) se sitúa en la mitad de la carrera*

$$\text{Posición límite derecha} = -51616,26/2 = -25808\text{pulos}$$

$$\text{STANDS\_EXCHANGER\_CALIB\_IMP\_MAX} = -25808.$$

7. Parada del motor sin rampa de desaceleración.
8. Se borran todas las variables de la estructura del filtro pid y del quadramp.
9. Se reajustan los variables de las estructuras del filtro pid y del quadramp a los valores ajustados para el programa. Los del PID fueron guardados en unas variables temporales en esta función.
10. Se actualiza la variable de estado de la estructura del carro horizontal “stands\_exchanger.calibrated” a “ON”, indicando que ya se realizó correctamente la calibración.

- void stands\_exchanger\_set\_position(int32\_t position\_mm)  
Función de ajuste de una nueva consigna de posición para el carro horizontal, el valor tiene que ser introducido en milímetros, en el sistema de referencia del sistema.

*Parámetros:*

- Posición en milímetros. El margen máximo está limitado desde 139 a -139.

*Funcionamiento interno:*

1. Satura el valor de posición al máximo y al mínimo:  
“STANDS\_EXCHANGER\_POSITION\_MAX\_mm”, fijado a 139.  
“STANDS\_EXCHANGER\_POSITION\_MIN\_mm”, fijado a -139.

2. Carga el nuevo valor de consigna, con previa conversión a pulsos.
3. Desactiva la variable de estado “stands\_exchanger.blocking” a OFF.

- **int32\_t stands\_exchanger\_get\_position(void)**

Función que devuelve la posición actual del carro en milímetros, en el sistema de coordenadas definido de usuario.

*Retorna:* Posición en milímetros, valor entre 139 y -139.

- **uint8\_t stands\_exchanger\_test\_traj\_end(void)**

Esta función comprueba si se da por finalizado el movimiento en el carro horizontal.

*Retorna:* Un código de estado de movimiento. Ver Tabla 3-4.

Valor	Macro	Significado
0	“no codificado”	Movimiento no finalizado
1	END_TRAJ	Si la consigna filtrada ya es la consigna de origen, y el error entre la posición alcanzada y la consigna filtrada es menor de 100 (Distancia a destino menor de 0,53 mm)
8	END_BLOCKING	Motor bloqueado

Tabla 3-4 Bits de código de estado de conclusión de movimiento de carro horizontal

*Funcionamiento interno:*

1. Si la consigna filtrada ya es la consigna de origen, y el error entre la posición alcanzada y la consigna filtrada es menor de 100 (0,53 mm). Se sale de la función retornando END\_TRAJ.
2. Si se produjo bloqueo, parada sin rampa de desaceleración, reseteo de las variables del filtro PID y del detector de bloqueo y se activa el flag “stands\_exchanger.blocking” a ON. Por último se sale de la función retornando END\_BLOCKING.
3. Si no se dio ninguna de las anteriores situaciones se retorna 0, indicando que el movimiento está en curso.

- **uint8\_t stands\_exchanger\_wait\_end(void)**

Función que invoca internamente a la función “stands\_exchanger\_test\_traj\_end()”. Tiene la misma funcionalidad. Añade el registro de un mensaje cuando se finaliza el movimiento por cualquiera de las causas posibles.

*Retorna:* Un código de estado de movimiento. Ver Tabla 3-4.

### 3.17.3. Función de control de bandeja de palomitas

- **int8\_t popcorn\_tray\_set\_mode(popcorn\_tray\_t \*popcorn\_tray, uint8\_t mode, int16\_t pos\_offset)**

Función de control de posicionamiento de la bandeja de palomitas. Solo se han definido dos posiciones que son indicadas mediante mode. Es posible definir un offset en pulsos de servo que será aplicado a la posición solicitada. Es ajustada la señal pwm del servo y sale de la función.

*Parámetros:*

- Dirección de la estructura de la bandeja de las palomitas.
- Código de la posición de desplazamiento. Se han codificado las siguientes:  
“POPCORN\_TRAY\_MODE\_OPEN” 0 Puerta completamente abierta.  
“POPCORN\_TRAY\_MODE\_CLOSE” 1 Puerta completamente cerrada.

*Retorna:* El valor de 0 con ejecución correcta y “-1” en caso de error

### 3.17.4. Función de control de pinzas de elevadores de stands

- **int8\_t stands\_clamp\_set\_mode(stands\_clamp\_t \*stands\_clamp, uint8\_t mode, int16\_t pos\_offset)**

Función de control de posicionamiento de los dos sistemas de pinzas de stands de ambos elevadores. La función recibe la dirección de la estructura de la pinza sobre la que se dirigirá el movimiento, en esta estructura esta identificado si la pinza corresponde al elevador derecho o al izquierdo.

Se han codificado tres posiciones definidas. Es posible definir un offset en pulsos de servo que será aplicado a la posición solicitada.

Es ajustada la señal pwm del servo analógico y se sale de la función.

*Parámetros:*

- Dirección de la estructura de la pinza de stands. En el programa hay declaradas dos estructuras, una por lado.
- Código de la posición de desplazamiento. Se han codificado las siguientes:  
“STANDS\_CLAMP\_MODE\_FULL\_OPEN” 0 Pinzas en posición de máximaapertura  
“STANDS\_CLAMP\_MODE\_OPEN” 1 Apertura mínima, stands liberados  
“STANDS\_CLAMP\_MODE\_CLOSE“ 2 Pinzas cerradas, stand sujetado.

*Retorna:* El valor de 0 con ejecución correcta y “-1” en caso de error.

### 3.17.5. Funciones de control de puertas de liberación de torres de stands

- **int8\_t stands\_tower\_clamps\_set\_mode(stands\_tower\_clamps\_t \*stands\_tower\_clamps, uint8\_t mode, int16\_t pos\_offset)**

Función de control de posicionamiento del sistema de puertas de torres de stands. Este sistema está compuesto por dos servos AX12 que deben de funcionar siempre con movimientos simultáneos y en este caso particular

también completamente simétricos. La función ajusta la posición destino en el registro de posición de destino de los servos AX12 y retorna 0.

La función recibe la dirección de la estructura de datos asociada al sistema, las posiciones codificadas que en este caso son tres y el offset en pulsos de sistema del servo que se añadirá a la posición seleccionada.

Esta función sigue la estructura del diagrama de flujo del Diagrama 3-6 *Diagrama de flujo "tipo" de funciones de control de actuadores*.

*Parámetros:*

- Dirección de la estructura del sistema de puertas de las torres de stands. En el programa hay declaradas solo una estructura de este tipo.
- Código de la posición de desplazamiento. Se han codificado las siguientes:  
“STANDS\_TOWER\_CLAMPS\_MODE\_UNLOCK\_LEFT” 0 Puerta de foco izquierdo abierta.  
“STANDS\_TOWER\_CLAMPS\_MODE\_LOCK” 1 Cerradas ambas puertas.  
“STANDS\_TOWER\_CLAMPS\_MODE\_UNLOCK\_RIGHT” 2 Puerta de foco derecho abierta.

*Retorna:* El valor de 0 con ejecución correcta y “-1” en caso de error.

- **uint8\_t stands\_tower\_clamps\_test\_traj\_end(stands\_tower\_clamps\_t \*stands\_tower\_clamps)**

Función que comprueba si el movimiento en los servos AX12 de la estructura de datos del sistema de puertas de torres de stands se pueda considerar finalizado. Esta función accede al estado de ambos servos mediante las funciones “ax12\_test\_traj\_end ()” que devuelven en estado de ejecución sin esperar a la finalización del movimiento. En la llamada a esta función han sido habilitados los flag “END\_NEAR” y “END\_TRAJ” por lo que todos los posibles valores devueltos se encuentran codificados en la Tabla 3-3 *Bits de código de estado de conclusión de movimiento de servos* Tabla 3-3.

Es posible conocer el estado de ambos servos gracias a que el valor de cada flag ha sido codificado en un bit único y el valor devuelto por esta función es el valor devuelto para ambos servos por separado unidos en único byte mediante una operación “OR” a nivel de bits. Aunque este sistema no nos va a permitir conocer si ambos servos terminaron su movimiento, si permitirá conocer si alguno de los dos se bloqueó.

Se considerara que el movimiento ha concluido cuando uno de los servos haya terminado su movimiento.

*Parámetros:*

- Dirección de la estructura de datos del sistema de puertas de torres de stands sobre la que se aplica el movimiento.

*Retorno:* uno de los valores de la Tabla 3-3, donde se codifica la posición y estado actual.

- **uint8\_t stands\_tower\_clamps\_wait\_end(stands\_tower\_clamps\_t \*stands\_tower\_clamps)**

Función que espera a que el movimiento en los servos AX12 de la estructura de datos del sistema de puertas de torres de stands se pueda considerar finalizado o se supere el triple del tiempo calculado para el movimiento programado. Esta función accede al estado de ambos servos mediante las funciones “ax12\_wait\_traj\_end( )” que bloquea la ejecución hasta que se de uno de los dos casos descritos. En la llamada a esta función han sido habilitados los flag “END\_NEAR” y “END\_TRAJ” por lo que todos los posibles valores devueltos se encuentran codificados en la Tabla 3-3.

Es posible conocer el estado de ambos servos gracias a que el valor de cada flag ha sido codificado en un bit único y el valor devuelto por esta función es el valor devuelto para ambos servos por separado unidos en único byte mediante una operación “OR” a nivel de bits. Aunque este sistema no nos va a permitir conocer si ambos servos terminaron su movimiento, si permitirá conocer si alguno de los dos se bloqueó.

No se retorna de esta función hasta que han concluido los movimientos de ambos servos o han quedado bloqueados.

*Parámetros:*

- Dirección de la estructura de datos del sistema de puertas de torres de stands sobre la que se aplica el movimiento.

*Retorno:* uno de los valores de la Tabla 3-3, donde se codifica la posición y estado actual.



## 4. Esquemas eléctricos

---

Todos los esquemas eléctricos han sido incluidos en la documentación entregada junto con el proyecto en el directorio:

[ESQUEMAS TARJETAS\mainboard\\_r15\\_eurobot2015\\_r1.pdf](#)

El listado completo incluido es el siguiente:

- Power Supply
- Main Microcontroller
- Slave Microcontroller
- Analog Sensors
- Digital Sensors
- Servo Motors
- DC Motor y Power Driving
- Brushless Motor
- Bluetooth and RS232



---

## BIBLIOGRAFIA

- BEA TRANSMISION. Scelta del passo delle cinghie dentada. En *Correas dentadas paso metrico serie "T"* (pág. 7).
- Eurobot. (2014). Imagen E2015\_Sketchup\_0001.jpg. En *Règlement Eurobot EN 2015* (pág. 1).
- Eurobot. (2014). *Normativa Eurobot 2015*.
- Eurobot. (2014). Playing area view from top. En *Règlement Eurobot EN 2015* (pág. 32).
- Hubrich, H. (2006). Characteristic diagram BG 40 x 50 24V. En *Q5 Katalog BG 01\_06* (pág. 1).
- International Rectifier. (2004). *PD-91645A*. Recuperado el 2015, de <http://www.irf.com/product-info/datasheets/data/irf7389.pdf>
- Maxon. (s.f.). *Pin allocation*. Obtenido de Encoder Enc 22 100 CPT, 2 Channels: [http://www.maxonmotor.nl/medias/sys\\_master/root/8816813965342/15-361-EN.pdf](http://www.maxonmotor.nl/medias/sys_master/root/8816813965342/15-361-EN.pdf)
- Microb Technology, Olivier MATZ. (2010). *wiki.droids-corp.org*. Recuperado el 2015, de <http://wiki.droids-corp.org/articles/a/v/e/Aversive.html>
- Microchip Technology Inc. (2006). *TC4426/TC4427/TC4428 1.5A Dual High-Speed Power MOSFET Drivers*. Recuperado el 2015
- Omron. (2005). *NPN Output circuit*. Recuperado el 2015, de E3Z Datasheet: <http://www.farnell.com/datasheets/1493598.pdf>
- Robotis. (2006). Recuperado el 2015, de <http://hackerspace.cs.rutgers.edu/library/Bioloid/doc/AX-12.pdf>
- User´s Manual Dynamixel AX-12. (2006). *Manual Dynamixel AX-12*. Recuperado el 2015, de <http://hackerspace.cs.rutgers.edu/library/Bioloid/doc/AX-12.pdf>
- Wikipedia. (s.f.). *Wikipedia*. Recuperado el 2015, de <https://es.wikipedia.org/wiki/Odometría>
- Zayas Figueras, E. y. (2008). *Departamento de Ingenieria Mecánica Universidad Politécnica de Cataluña*. Recuperado el 2015, de <http://em.upc.edu/ca/docencia/estudis-degrau/etseib/TFTM/quadern-de-treball/versio-castellana/leccion-4>