

Proyecto Final
Robot recolector de residuos
Diseño, implementación y construcción física

Guillermo Campelo
Juan Ignacio Goñi
Diego Nul

24 de junio de 2010

Resumen

Índice

1. Requerimientos	5
2. Ideas de implementación	5
2.1. Locomoción	5
2.2. Sensado del entorno	5
2.3. Controlador	5
2.4. Método de recolección	5
3. Actuadores	5
3.1. Motores de continua	5
3.1.1. Características	5
3.1.2. Circuito de control	6
3.1.3. Rutinas de control	7
3.2. Servo motores	8
3.2.1. Circuito de control	8
3.2.2. Rutinas de control	9
4. Sensado	9
4.1. Telémetros infrarrojos	9
4.1.1. Características	9
4.1.2. Circuito de control	9
4.1.3. Rutinas de control	11
4.2. Sensor de distancia por ultrasonido	11
4.2.1. Características	11
4.2.2. Circuito de control	11
4.2.3. Rutinas de control	12
4.3. Sensor reflectivo de piso	13
4.3.1. Características	13
4.3.2. Circuito de control	14
4.3.3. Rutinas de control	14
4.4. Encoders	14
4.4.1. Características	14
4.4.2. Circuito de control	15
4.4.3. Rutinas de control	15
4.5. Sensado de la batería	15
4.6. Consumo del motor	15
4.6.1. Pulsador u otro dispositivo disparador	16
4.6.2. Rutinas de control	17
5. Controladores	17
5.1. Netbook	17
5.2. Microcontrolador	17
5.2.1. Características	17
5.2.2. Módulos internos	18
5.2.3. Programación del firmware	19

6. Comunicación	19
6.1. Conectividad entre módulos	19
6.2. Protocolo de comunicación	19
6.2.1. Características básicas	19
6.2.2. Comandos comunes	19
6.2.3. Comandos específicos	19
6.2.4. Estadísticas	20
7. Placas controladoras	20
7.1. Placa genérica	20
7.1.1. Características principales	20
7.1.2. Módulo de comunicación	20
7.1.3. Alimentación de la placa	20
7.1.4. Configuración	20
7.1.5. Esquemático	20
7.1.6. Circuito	20
7.1.7. Código básico	20
7.1.8. Posibles extensiones	20
7.2. Placa controladora de motores DC	21
7.2.1. Características principales	21
7.2.2. Módulo de comunicación	21
7.2.3. Alimentación de la placa	21
7.2.4. Configuración	21
7.2.5. Esquemático	21
7.2.6. Circuito	21
7.2.7. Código básico	21
7.2.8. Posibles extensiones	21
7.3. Placas de sensado	21
7.3.1. Características principales	22
7.3.2. Módulo de comunicación	22
7.3.3. Alimentación de la placa	22
7.3.4. Configuración	22
7.3.5. Esquemático	22
7.3.6. Circuito	22
7.3.7. Código básico	22
7.3.8. Posibles extensiones	22
7.4. Placa controladora de servo motores	22
7.4.1. Características principales	22
7.4.2. Módulo de comunicación	22
7.4.3. Alimentación de la placa	23
7.4.4. Configuración	23
7.4.5. Esquemático	23
7.4.6. Circuito	23
7.4.7. Código básico	23
7.4.8. Posibles extensiones	23

8. Armado del prototipo	23
8.1. Diseño	23
8.2. Características	23
8.3. Desarme	23
8.4. Costo y proveedores	23
A. Primer apéndice Hardware	24
B. Segundo apéndice Hardware	24

1. Requerimientos

2. Ideas de implementación

2.1. Locomoción

distintos tipos de locomoción que tuvimos en cuenta y porque elegimos este

2.2. Sensado del entorno

distintos tipos de sensores disponibles y porque elegimos estos

2.3. Controlador

distintas formas de diagramar la forma de control, que tipo de controladores necesitamos, en cuales pensamos, con cuales nos quedamos

2.4. Método de recolección

distintos metodos que se nos ocurrieron

3. Actuadores

En nuestro caso, los motores son la principal forma en que el robot puede interactuar activamente con el ambiente que lo rodea. Cada una de las tareas que debíamos realizar requería de actuadores acordes.

Estas cuestiones son las que analizamos en este apartado.

3.1. Motores de continua

Para la tracción principal de las ruedas necesitabamos motores que tuvieran el torque necesario para mover el robot, pero que pudieramos medir y controlar la velocidad era la principal necesidad. Para esta tarea utilizamos motores de continua con caja reductora y encoder. Con dos de estos motores logramos poder garantizar una velocidad determinada en las ruedas, controlar de la cantidad de movimiento en forma independiente en cada rueda y entre otras cosas conocer la cantidad de las vueltas dadas por cada una de las ruedas.

3.1.1. Características

Los motores que elegimos son de la marca Ignis ¹ modelo *MR-2FA* con características expresadas en la tabla 1, están provistos de una caja reductora, poseen un encoder de 4 estados por vuelta en el eje del motor y un sensor de efecto de campo para determinar una vuelta en la salida de la caja reductora.

La caja reductora provee una relación de 94 vueltas del motor por cada 1 vuelta del eje de salida de la caja.

En la figura 1 mostramos las dimensiones exteriores del motor.

¹<http://www.ignis.com.ar>

Característica	Unidad	Mínimo	Nominal	Máximo
Tensión	V	8	9	12
Corriente	A	0.6	1.2	2.4
Velocidad	RPM	1	60	60
Aceleración	$1/s^2$	0.1	0.1	0.5
Torque	kgf*cm	0	1.2	6.4

Cuadro 1: Características del motor Ignis MR-2FA.

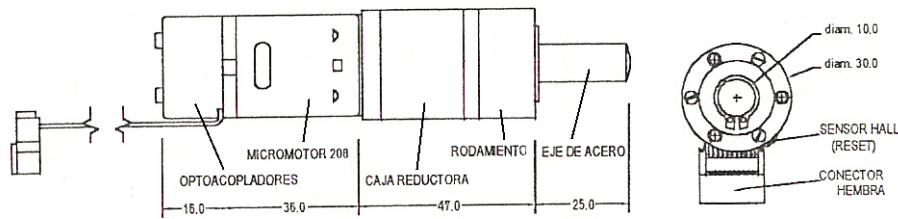


Figura 1: Vista lateral y frontal del motor Ignis MR-2FA.

Una ventaja que encontramos en este modelo es que ya trae el encoder integrado aunque su resolución podría haber sido mayor. Los encoders los explicamos más en detalle en la sección 4.4.

3.1.2. Circuito de control

Para alimentar y poder controlar los motores elegimos el driver *L298* de la marca ST ². Internamente tiene dos puentes H puentesables y puede soportar hasta 4A. Ideal para estos motores u otros que se elijan en el futuro.

La principal función del driver era proveer de la corriente y voltaje necesarios para el funcionamiento del motor, pero la configuración del puente H nos dio la posibilidad de, con una lógica simple, determinar el sentido de la corriente y potencia que recibía el motor.

Este integrado admite el puenteo de la salida aumentando así la corriente que pueda circular por él. Para hacer esto, conectamos las salidas *Out1* y *Out4* por un lado y por el otro las salidas *Out2* y *Out3*. De igual forma los pines de habilitación *EnA* y *EnB*, luego la entrada *In1* con la *In4* por un lado y por el otro la *In2* con la *In3*.

De esta forma, se controla con sólo 3 cables, uno de habilitación y otros 2 de *Input* que determinan la polarización de los transistores internos y por ende, el sentido de giro del motor. En la tabla 2 mostramos la tabla de verdad para los pines de control. En la figura 2 mostramos el diagrama interno del integrado.

Para determinar la potencia que recibiría el motor usamos el módulo de *PWM* del microcontrolador, que explicamos más en detalle en la sección 5.2.2. Variando el ancho del pulso sobre el pin de habilitación del driver determinamos la cantidad de tiempo que el motor recibe tensión, lo cual se traduce en la

²<http://www.st.com>

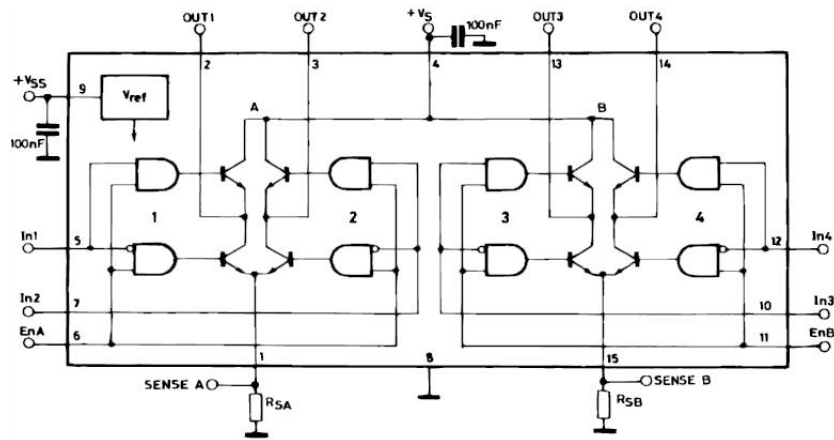


Figura 2: Diagrama interno del driver L298.

Enable	Input 1	Input 2	Función
H	H	L	Sentido horario
H	L	H	Sentido anti-horario
H	L	L	Motor frenado
H	H	H	Motor frenado
L	X	X	Motor libre

Cuadro 2: Tabla de verdad para el control del driver *L298*. Donde H es estado alto, L estado bajo y X cualquier estado.

potencia que este tiene para realizar el movimiento. Cuanto mayor es el tiempo en estado alto del pulso, mayor la potencia.

Para contrarrestar la corriente negativa en las salidas del driver usamos los diodos *FR304* que cumplen con las especificaciones del driver con 150ns de tiempo de recuperación y una corriente de 3A.

El consumo del motor lo medimos mediante los pines de sensado en el driver, conectado a masa por una resistencia y al módulo de *ADC* del microcontrolador, que explicamos en la sección 5.2.2. Conociendo el valor de la resistencia y el valor leído por el módulo de *ADC*, pudimos determinar cuanta corriente que circulaba por la resistencia y por ende la corriente consumida por el motor.

Para controlar la velocidad del motor usamos 1 de los 2 encoders que trae. Conectados como entrada para el módulo de *Timer* del microcontrolador, en configuración de contador, que explicamos más en detalle en la sección 5.2.2. Usando otro *Timer* para tener una base de tiempo fija y con el valor del contador pudimos determinar y controlar la velocidad de las ruedas.

En la sección 7.2 explicamos el desarrollo de la placa controladora de estos motores.

3.1.3. Rutinas de control

Desde el punto de vista del código, tuvimos que desarrollar las rutinas necesarias para el manejo de los motores según las instrucciones del controlador prin-

Característica	Unidad	Valor
Torque	kg	6,5
Velocidad	segundos/grado	$\frac{0,16}{60}$
Voltaje	V	4,8 a 6
Delay máximo	μs	4
Dimensiones	mm	40x20x38
Peso	g	39

Cuadro 3: Características del servo HX5010.

cial.

Configuramos a uno de los timers internos del microcontrolador para que genere una interrupción cada 6,25ms, la cual usamos para realizar chequeos y ejercer control sobre el motor. Verificamos el consumo del motor para evitar sobrecargar el circuito y los motores ante un posible atasco de las ruedas. También actualizamos el acumulado de vueltas realizadas por el motor para la odometría.

Cada 200ms, o sea 32 interrupciones, tomamos la cantidad de cuentas del encoder, corregimos la velocidad de giro del motor ajustando el ancho del pulso generado por el PWM. Luego borramos el contador y esperamos otros 200ms.

3.2. Servo motores

Para el movimiento de las partes del módulo de recolección, una cámara con paneo y giro o un sensor de ultrasonido colocado en la parte superior haciendo las veces de radar, pensamos en el uso de servo motores. La principal característica de estos actuadores es que nosotros sólo debemos indicar el ángulo al que queremos que este el eje del motor y este se coloca automáticamente. El ángulo de trabajo va desde 0° a 180° y algunos llegan hasta los 200°.

Aunque no fue implementado el ningún mecanismo que requiriera el uso de estos motores, explicamos en este apartado el trabajo realizado en torno a este tipo de actuadores. En la sección 7.4 explicamos el diseño de las placas que los controlan.

El servo de prueba que utilizamos para el desarrollo es el modelo *HX5010* de la marca Hextronik ³ con características que expresamos en la tabla 3

3.2.1. Circuito de control

La alimentación y consumo depende del modelo específico, variando también el torque que posee el servo.

No es necesario el uso de un driver para manejarlos, simplemente con la alimentación y una señal con el ángulo es suficiente. La forma de comunicar el ángulo varía entre los distintos servos y fabricantes. Hay servos analógicos y servos digitales. En los primeros la posición se determina mediante un voltaje que varía según cierto rango y si es digital, se setea mediante el ancho de un pulso que tiene un tiempo mínimo y máximo para mapear los ángulos mínimo y máximo respectivamente.

³www.hextronik.com/

Dentro del modo de uso, podemos hacer que queden sueltos o que se queden fijos en cierta posición indicando, de forma continua, el valor del ángulo requerido. La frecuencia a la que se debe setear la posición depende el modelo.

3.2.2. Rutinas de control

Debido a que pensamos usar servos digitales y por lo menos íbamos a necesitar 3 servos, necesitábamos contar con varios módulos de PWM. Como sólo disponíamos de 1, decidimos realizar la misma función pero por software.

Usamos el timer de 16bits del microcontrolador configurado con el clock interno como medida del tiempo para crear 5 salidas con pulsos que varían el ancho en forma independiente cada una. Definimos un ancho mínimo y máximo, pudiendo configurar pasos intermedios de 1° (aproximadamente $69,4\mu s$).

4. Sensado

En este apartado explicamos detalladamente cada uno de los sensores que utilizamos para realizar tanto las mediciones externas como las internas al robot. Analizamos las ventajas de cada uno, problemas que encontramos y sus soluciones.

En la sección 7.3 explicamos el diseño y construcción de las placas que controlan todos los sensores del robot.

4.1. Telémetros infrarrojos

El principio de funcionamiento de estos sensores es mediante un haz de luz infrarroja que es emitido hacia el objetivo, el cual es reflejado y captado a través de un lente por un sensor de posición relativa en el interior del sensor. En base a esta medición se calcula la distancia entre el sensor y el objeto reflectivo que se encuentra frente a él.

4.1.1. Características

medidas, tiempo de muestreo, tipo de salida, rangos de distancia, rangos de voltaje, distancia vs voltaje

Los telémetros infrarrojos que elegimos son de la marca Sharp ⁴, modelo *GP2D120*. En la tabla 4 detallamos los valores característicos del modelo.

Este tipo de sensores tienen un retardo de aproximadamente $43,1ms$ durante el cual la lectura que se realiza no es confiable y luego las nuevas lecturas se hacen en ventanas de aproximadamente el mismo tiempo. En la figura 3 mostramos el diagrama de tiempos.

4.1.2. Circuito de control

alimentación, conmutación (transistor, estado de habilitación: 0), conexión con el micro, salida del sensor, módulo ADC, muestreo, capacitor para alimentación, circuito mínimo (diagrama)

⁴<http://sharp-world.com/products/device>

Característica	Unidad	Valor
Rango máximo	cm	30
Rango mínimo	cm	4
Tensión para la máxima distancia	V	1.95
Tensión para la mínima distancia	V	2.55
Tensión de alimentación	V	5
Consumo máximo	mA	50

Cuadro 4: Características del sensor de distancia por ultrasonido SRF05.

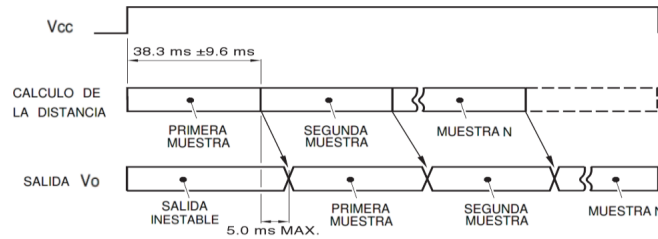


Figura 3: Diagrama de tiempos del sensor GP2D120.

No necesitábamos un driver para manejar los telémetros, pero decidimos usar un transistor para poder habilitarlos o no, de otra manera el sensor estaba tomando mediciones continuamente provocando un consumo de batería innecesario. De esta forma sólo se enciende cuando se va a utilizar.

El transistor que utilizamos es un conmutador y amplificador de uso general, el *BC327* y por ser de tipo PNP se exita con un estado bajo, por lo que la lógica de conmutación está negada.

El tipo de salida de este modelo de telémetros es analógica y están conectadas al módulo *ADC* del microcontrolador. En la figura 4 mostramos la tabla de conversión entre voltaje de salida y distancia al objeto, y en la figura 5 mostramos el ángulo de apertura de la zona de detección según la distancia al objetivo.

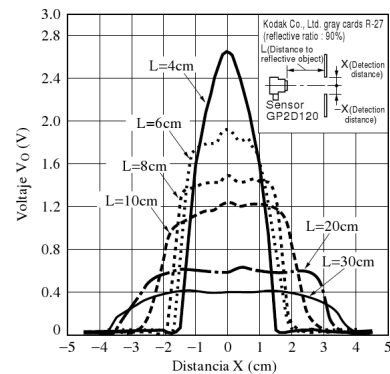
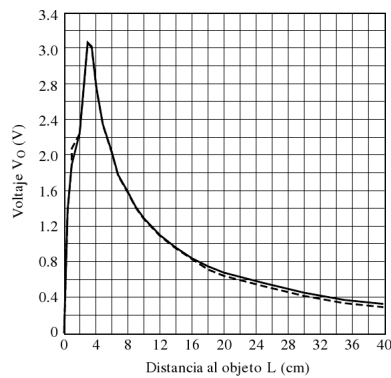


Figura 4: Voltaje de salida según la distancia al objeto del telémetro GP2D120. Figura 5: Ángulo de apertura según la distancia del telémetro GP2D120.

4.1.3. Rutinas de control

Controlar los telémetros es relativamente sencillo. Usamos el timer de 16bits del microcontrolador configurado con el clock interno para determinar el tiempo en el cual debíamos tomar las muestras con el ADC y simplemente realizamos un promedio entre ellas para obtener la distancia al objetivo.

Para hacer esto desarrollamos una pequeña máquina de estados que controla y maneja los tiempos para tomar las muestras que explicamos más en detalle en la sección 7.3.7.

4.2. Sensor de distancia por ultrasonido

Estos sensores de distancia se basan en la velocidad del sonido para calcular la distancia al objetivo. Genera un tren de 8 pulsos ultrasónicos y luego se espera como respuesta, el mismo tren de pulsos que debería haber rebotado contra el objetivo. En base a la diferencia de tiempo entre la emisión del tren de pulsos y la respuesta, se calcula la distancia a la que se encuentra el objetivo.

4.2.1. Características

El sensor de distancia por ultrasonido que elegimos es el modelo *SRF05* de la marca Devantech Ltd ⁵. Esta versión mejorada del modelo *SRF04*, aumenta el rango de detección y mejora el modo de control y lectura de los datos, permitiendo hacerlo mediante un único pin.

La distancia medida mediante el tren de pulsos es codificada linealmente en el ancho de un pulso que varía de $100\mu s$ a $25ms$. Si dentro del rango de detección no se encuentra ningún objeto, el pulso tendrá un ancho de $30ms$.

En la tabla 5 detallamos las características del sensor *SRF05* y en la figura 6 mostramos el haz ultrasónico del sensor.

Característica	Unidad	Valor
Tensión de alimentación	<i>V</i>	5
Corriente	<i>mA</i>	4
Frecuencia de trabajo	<i>KHz</i>	40
Rango máximo	<i>cm</i>	400
Rango mínimo	<i>cm</i>	1.7
Duración mínima del pulso de disparo	μs	10
Duración del pulso eco de salida	μs	100 - 25000
Tiempo mínimo de espera entre mediciones	<i>ms</i>	50
Dimensiones	<i>mm</i>	43x23x40

Cuadro 5: Características del sensor SRF05.

4.2.2. Circuito de control

No necesitamos de un driver para manejar al sensor ya que lo conectamos directo a los 5v de la placa. El pin de *Mode* lo dejamos en estado bajo para

⁵<http://www.robot-electronics.co.uk/>

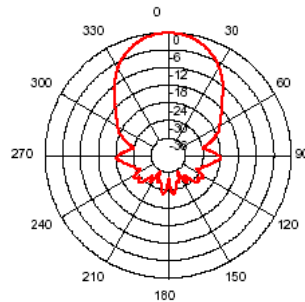


Figura 6: Haz ultrasónico del sensor SRF05.

indicar que debe funcionar bajo el nuevo modo y no en compatibilidad con el *SRF04*.

En el pin de *TRIGGER* sólo generamos un pulso de al menos $10\mu s$ para desencadenar en la lectura de la distancia al objetivo. El sensor nos asegura que no generará el pulso de respuesta hasta pasados los $700\mu s$ desde pasado el pulso de trigger. En la figura 7 mostramos el diagrama de tiempos.

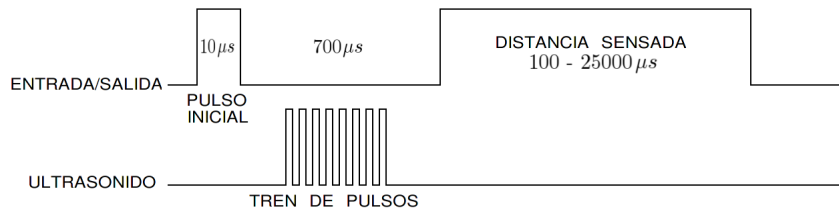


Figura 7: Diagrama de tiempos del sensor SRF05.

Para evitar que el rebote de otros sensores o sensados anteriores influya en la lectura, se debe esperar un mínimo de $50ms$ antes de generar otro medición.

4.2.3. Rutinas de control

Usamos uno de los pines con interrupción externa en el que conectamos el pin de *TRIGGER* y el timer de 16bits del microcontrolador configurado con el clock interno.

Para realizar la medición, generamos un pulso de $15\mu s$ para asegurarnos el disparo del sensor y cambiamos el modo del pin a entrada con interrupción ante un flanco ascendente. Cuando salta la interrupción significa que comienza al pulso con la distancia codificada en su ancho, por lo que tomamos una muestra del timer y configuramos al pin para que genere ahora una interrupción ante un flanco descendente. Cuando salte la próxima vez la interrupción, será porque termino el pulso con la medición, por lo que sólo debemos hacer la resta entre el valor actual del timer y la muestra que tomamos al principio para conocer la distancia a la que se encuentra el objetivo.

Un tiempo obtenido mayor a los $25ms$ indica que no se detectó ningún objeto dentro del rango del sensor.

4.3. Sensor reflectivo de piso

Estos sensores opticos reflectivos emiten luz infrarroja y captan el nivel de luz reflejada sobre la superficie a sensar como mostramos en la figura 9. En la figura 8 mostramos las dimensiones del sensor. La intensidad de luz captada depende de la distancia al objetivo y del color y nivel de reflectividad de la superficie. Es por esto que usamos estos sensores para identificar una línea en el piso.

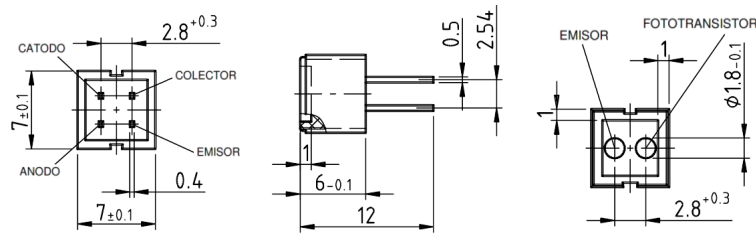


Figura 8: Medidas en milímetros del sensor CNY70.

4.3.1. Características

Los sensores que elegimos para son del modelo *CNY70* de la marca Vishay Semiconductor ⁶.

El rango efectivo de sensado ronda los $3mm$ de distancia aunque, con un incremento en la corriente que circula por el emisor se puede llegar a una distancia mayor a la recomendada por el fabricante y que nos permita un uso más acorde al proyecto. El emisor soporta un pulso de hasta $3A$ por un tiempo menor o igual a $10\mu s$.

En la figura 10 mostramos la corriente que circula por el colector del fototransistor en base a la distancia al objeto medido.

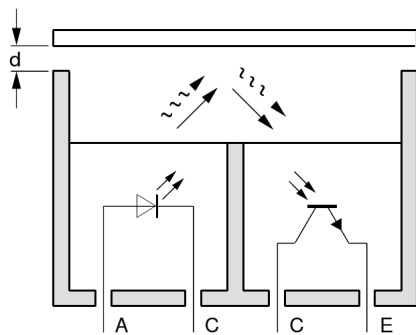


Figura 9: Principio de funcionamiento reflectivo del sensor CNY70.

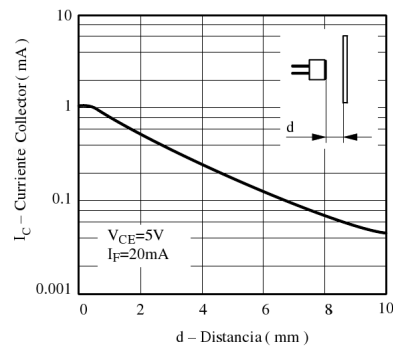


Figura 10: Corriente en el colector según la distancia del sensor CNY70.

⁶<http://www.vishay.com/>

4.3.2. Circuito de control

De igual forma que en los telémetros utilizamos un transistor como habilitación de la tensión de alimentación en el sensor. Esto nos dio la posibilidad de encenderlo y apagarlo a la hora de tomar las muestras de la luz reflejada por piso o la línea. Nuevamente la lógica de habilitación es invertida por tratarse de un transistor *BC327*.

Agragamos una resistencia para limitar la corriente que circulaba por el emisor y otra como pull-up en el emisor del fototransistor que a su vez, conectamos al módulo de *ADC* del microcontrolador para efectuar las mediciones.

4.3.3. Rutinas de control

El código que desarrollamos para obtener las muestras de estos sensores es sencillo, simplemente debemos habilitar el transistor que alimenta al sensor con un estado lógico bajo, tomar al menos 4 muestras y promediarlas para tener un valor adecuado del nivel de luz reflejado por la superficie. Deshabilitamos el sensor y enviamos el valor.

Debido al circuito que armamos con un nivel alto de reflexión leemos un valor bajo en el conversor analógico digital y con un nivel bajo de luz, un valor alto.

4.4. Encoders

Los encoders son sensores que convierten una posición lineal o angular en señal eléctrica o pulsos. Pueden determinar una posición de forma absoluta o simplemente informar que hubo un movimiento. El método de sensado y la resolución del ángulo de giro que detectan varía según el modelo.

4.4.1. Características

Los motores *MR-2FA* tienen encoders de cuadratura conectados al eje, previo a la caja reductora. Estos encoders son de tipo fotoeléctricos, están dispuestos a 135° uno del otro y marcan 4 estados por cada vuelta del motor. Están colocados así para poder conocer el sentido de giro midiendo la secuencia de estados de cada encoder. Adicionalmente el motor cuenta con un sensor de efecto Hall el cual nos permite detectar una revolución completa en el eje de salida de la caja reductora.

Nosotros como conocemos el sentido de giro del motor, pues lo determinamos con el puente H, necesitamos sólo uno de los dos encoders para conocer y controlar la velocidad a la que gira el rotor del motor.

A la tensión máxima los motores superan a las 320 cuentas por segundo, el mínimo y máximo recomendables son, para que podamos mantener un giro constante, 60 y 300 cuentas por segundo respectivamente. Estos cálculos son usando uno de los dos fototransistores del encoder.

Dependiendo el tamaño de las ruedas será la velocidad de final del robot. La relación de caja de 94 : 1 del motor.

4.4.2. Circuito de control

Conectamos una resistencia pull-up a 5V para la salida de sensado de los fototransistores y del sensor de efecto Hall. También incluimos un switch doble inversor para elegir cuál de los dos encoders usar. El punto común del switch está conectado al pin de entrada de clock externo de uno de los Timers del microcontrolador.

La alimentación de los encoders es directa y permanecen encendidos en todo momento.

4.4.3. Rutinas de control

Como explicamos en la sección 3.1.3, en cada interrupción del actualizamos el histórico de cuentas del motor adicionando o restando el último valor del contador.

También comparamos contra las cuentas esperadas por intervalo de tiempo que fueron determinadas desde el controlador principal para poder determinar si debemos incrementar o disminuir la potencia del motor y por ende la velocidad de las ruedas.

4.5. Sensado de la batería

El sensado del nivel de tensión en la batería lo hacemos mediante un divisor de tensión entre los polos de la batería. La salida del divisor es sensada de igual forma que los otros sensores, mediante el módulo de *ADC* del microcontrolador. En la figura 11 mostramos el diagrama del divisor de tensión.

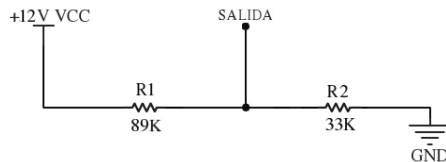


Figura 11: Divisor de tensión para el sensado de la batería.

En la tabla 6 mostramos las posibles tensiones en la batería y la tensión de salida en el divisor. También incluimos el valor aproximado para el ADC con tensión de referencia a 5V que leería la salida del divisor. El rango de voltajes que analizamos es teniendo en cuenta la posibilidad de efectuar mediciones durante la carga de la batería y sabiendo que con una tensión menor a 5V la lógica del prototipo que construimos comenzaría a fallar.

La conexión es simple, los cables de entrada del divisor los conectamos a los polos de la batería y la salida al ADC. Luego sólo debemos realizar las lecturas en el ADC y promediarlas para poder realizar los cálculos de la tensión en la batería.

4.6. Consumo del motor

El consumo de los motores de corriente continua lo medimos leyendo el pin de sensado que se encuentra en el puente H que alimenta al motor. Lo que medimos con el módulo de *ADC* del microcontrolador es la tensión en este

Batería (V)	Salida (V)	Valor en el ADC
16	4.327	886
15	4.057	831
14	3.786	776
13	3.516	720
12	3.245	665
11	2.975	609
10	2.704	554
9	2.434	499
8	2.163	443
7	1.893	388
6	1.623	332
5	1.352	277

Cuadro 6: Tensión de la batería y la tensión de salida en el divisor.

pin. Esta depende de la caída de tensión en la resistencia conectada a masa y de la corriente que circula por el motor. Conociendo el valor de la resistencia podemos calcular el consumo del motor. En la tabla 7 comparamos el consumo en el motor, el voltaje sensado y la lectura en el ADC.

Tensión (V)	Consumo (A)	Valor en el ADC
0	0	0
0,09	0,19	18
0,18	0,38	36
0,27	0,57	55
0,36	0,76	73
0,45	0,95	92
0,54	1,14	110
0,63	1,34	129
0,72	1,53	147
0,81	1,72	165
0,90	1,91	184
0,99	2,10	202
1,08	2,29	221
1,17	2,48	239

Cuadro 7: Tabla comparativa para el consumo del motor.

4.6.1. Pulsador u otro dispositivo disparador

Ademas de los sensores que describimos, pensamos que nos podrían hacer falta pulsadores para controlar funciones simples por ejemplo saber cuando una parte de algún mecanismo llega a cierto punto o detectar finales de carrera de un servo o sin fin. También pueden ser otro tipo de sensores que generen un cambio de estado en el pin de sensado que se conecta al microcontrolador.

Agregamos la posibilidad de usarlos en las distintas placas como explicamos

en detalle en las secciones 7.3 y 7.4.

4.6.2. Rutinas de control

La lectura en el estado de los pulsadores puede ser bajo demanda con solo leer el estado del pin en el que estan conectados o puede ser ante una interrupción por cambio de estado. Estas cuestiones las tuvimos en cuenta a la hora de diseñar las placas.

5. Controladores

Todas las funciones del robot las debíamos controlar mediante algún tipo de dispositivo. Decidimos utilizar una netbook y microcontroladores para esta tarea. Estas cuestiones son las que explicamos en este capítulo.

5.1. Netbook

modelo, marca, características, para que se usa, sistema operativo y lenguaje de programación

5.2. Microcontrolador

Para el control de la velocidad de los motores, lectura de los encoders y los sensores de distancia usamos un microcontrolador. Nuestro diseño contempló la existencia de varios módulos con una o pocas funciones simples que se comunicaran entre ellos y con el controlador principal, en nuestro prototipo la netbook. La razón por la cual lo armamos así fue para simplificar cada placa controladora a nivel software y hardware.

Explicamos la comunicación entre los distintos controladores en la sección 6.

5.2.1. Características

El microcontrolador que elegimos para realizar las tareas de control, configuración y comunicación a bajo nivel es el *PIC16F88* de Microchip ⁷. Cuenta con una arquitectura de memoria del tipo Harvard, con una memoria *FLASH* para 4096 instrucciones de programa, una memoria *RAM* de 368 bytes y una memoria *EEPROM* de 256 bytes. Tiene un set de instrucciones básicas reducido y todas con el mismo tiempo de ejecución. En este apartado nombramos algunos de los principales periféricos incluidos en el microcontrolador y la utilidad dentro del proyecto que encontramos para ellos. Utilizamos con un cristal externo de 20MHz como clock.

El microcontrolador tiene 2 puertos de 8 entradas y salidas cada uno de tipo TTL y CMOS. Como mostramos en la figura 12 cada pin se encuentra multiplexado con uno o más periféricos internos.

⁷<http://www.microchip.com/>

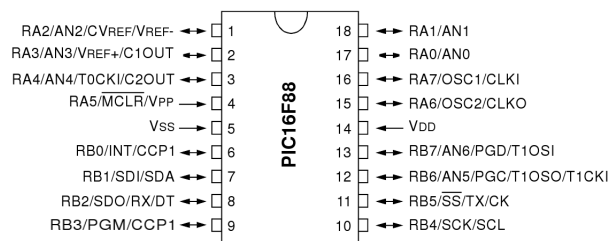


Figura 12: Diagrama del microcontrolador PIC16F88.

5.2.2. Módulos internos

Internamente el microcontrolador tiene una serie de periféricos que proveen de funciones extras y que utilizamos para lograr cumplir con las necesidades de nuestro proyecto. En la figura 13 mostramos los distintos módulos internos del microcontrolador.

Cuenta con 3 timers, 2 de 8bits (*TIMER0* y *TIMER2*) y 1 de 16bits (*TIMER1*). Podemos configurarlos para que tomen al clock del microcontrolador o que tomen una fuente externa de clock. También podemos aplicarles demultiplicadores que generan un clock de menor frecuencia al que se usa como entrada. Pueden ser etapas previas o posteriores al timer y nos dan gran flexibilidad de uso.

El *TIMER0* lo utilizamos para hacer control del tiempo en nuestros códigos. Conectados con el clock principal y configurados para que generen una interrupción al hacer overflow, obtenemos una buena medida del paso del tiempo.

El *TIMER1* configurado como fuente externa a la salida del encoder, lo usamos como contador de pasos para medir la velocidad del motor. También lo usamos como medición del tiempo para hacer las lecturas de los sensores. Elegimos a este timer ya que al ser de 16bits posee un mayor rango de valores y por lo tanto, podíamos medir un mayor lapso de tiempo o cuentas del encoder en cada caso.

El *TIMER2* lo usamos en conjunto con el módulo de *PWM* para determinar el ancho del pulso que habilita al puente H que provee de energía a los motores.

El módulo conversor analógico digital nos dio la posibilidad de medir tensiones analógicas como por ejemplo las salidas de los sensores, tensión en la batería o el consumo de los motores. Tiene 7 canales o pines distintos y pode-

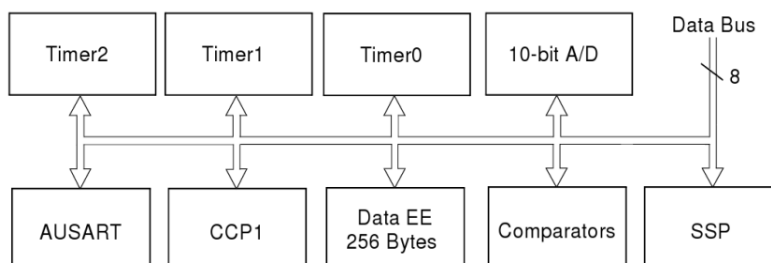


Figura 13: Módulos internos del microcontrolador PIC16F88.

mos configurarlo para que genere un valor de 8 o 10bits. También podemos determinar si se debe usar el valor de V_{cc} y GND como referencia o podemos proveer de forma externa de los voltajes de referencia para generar un rango de voltajes diferente y aumentar o disminuir así la resolución del conversor.

El módulo de *PWM* nos provee la posibilidad de generar pulsos continuos de un ancho determinado. En nuestro proyecto lo utilizamos como habilitación del puente H que alimenta a los motores variando así la potencia y por lo tanto, la velocidad final de las ruedas. Se utiliza en conjunto con el *TIMER2* seteando el prescaler y postscaler para determinar el ancho del estado alto y del estado bajo de los pulsos.

Gracias al módulo de *AUSART* podemos realizar la comunicación entre los distintos microcontroladores por hardware. Este periférico nos provee una comunicación sincrónica o asincrónica dependiendo de la configuración. Nosotros creamos la red de *Daisy Chain* sobre RS-232.

GIO EEPROM CCP SSP

5.2.3. Programación del firmware

pins de programacion, icd2, IDE, lenguaje, version

6. Comunicación

porque necesitamos comunicar los modulos, que necesidades hay, nivel de uso

6.1. Conectividad entre módulos

daisy chain, diagrama, montado sobre rs232, control de errores

6.2. Protocolo de comunicación

características necesarias en el protocolo, porque es importante, cosas que tuvimos en cuenta y decisiones, control de errores

6.2.1. Características básicas

formado por paquetes, formato basico del paquete (header), control de errores

6.2.2. Comandos comunes

contelo o listado de comandos comunes (en detalle o se van a un apendice)
- son pocos.

6.2.3. Comandos específicos

contelo o listado de comandos específicos segun el tipo de placa (referencia a un apendice con cada uno explicado)

6.2.4. Estadísticas

análisis de paquetes por segundo, bytes de datos vs bytes de header, retransmisiones, etc

7. Placas controladoras

porque tuvimos que diseñar nuestras propias placas, cosas que tuvimos en cuenta y decisiones tomadas, códigos fuente a los apéndice

7.1. Placa genérica

función de una placa genérica, porque fue armada, para que sirva

7.1.1. Características principales

testeo de nuevos módulos, testeo de la programación, sniffear la comunicación, futuras expansiones

7.1.2. Módulo de comunicación

explicación de la comunicación, igual en todas, switch de configuración, pines, fichas, nodos en la cadena, cables pc-placa y placa-placa, max232

7.1.3. Alimentación de la placa

tensión para la alimentación, circuito de la fuente, consumo máximo, voltaje mínimo de alimentación, alimentación de 5V directos

7.1.4. Configuración

configuración mínima de la placa, leds, comunicación, header de programación

7.1.5. Esquemático

esquemas de la placa

7.1.6. Circuito

circuito de la placa

7.1.7. Código básico

explicación de lo mínimo que debería tener para ser parte de la cadena de comunicación

7.1.8. Posibles extensiones

posibles extensiones a futuro de la placa - nuevos módulos de testeo o control o lectura muy básica de señales, pasar a montaje superficial los componentes, hacerla más chica

7.2. Placa controladora de motores DC

funcion de una placa controladora de motorDC, porque fue armada, para que sirve, porque hay 2, porque no esta en una sola

7.2.1. Características principales

principio de funcionamiento, como logra controlar la velocidad, como logra ser parte de la cadena, como logra sensar el consumo, controlar el motor, puente H, diodos, leds, VREF

7.2.2. Módulo de comunicación

se explico en el modulo generico, se agregan los comandos especificos y se puede explicar como se obtiene la informacion para dar las respuestas

7.2.3. Alimentación de la placa

se explico en el modulo generico, tension para la alimentacion para los motores, necesidad de masa unica como referencia, consumo aproximado de los motores

7.2.4. Configuración

configuracion de la placa, leds, comunicacion, header de programacion, switch de seleccion de encoder

7.2.5. Esquemático

esquematicos de la placa

7.2.6. Circuito

circuito de la placa

7.2.7. Código básico

explicacion de lo minimo que deberia tener para ser parte de la cadena de comunicacion, sensado y control de la velocidad de los motores

7.2.8. Posibles extensiones

unificar en una placa el control de mas de un motor, pasar a montaje superficial los componentes, hacerla mas chica

7.3. Placas de sensado

funcion de una placa de sensado, porque fue armada, para que sirve, que tipo de sensores puedo conectar, cuales son las posibles configuraciones, diferencias, sensado de la bateria

7.3.1. Características principales

principio de funcionamiento, como logra tomar las muestras de los sensores, como logra ser parte de la cadena, seteo de los tipos de sensores

7.3.2. Módulo de comunicación

se explico en el modulo generico, se agregan los comandos especificos y se puede explicar como se obtiene la informacion para dar las respuestas

7.3.3. Alimentación de la placa

se explico en el modulo generico

7.3.4. Configuración

configuracion de la placa, comunicacion, header de programacion

7.3.5. Esquemático

esquematicos de la placa

7.3.6. Circuito

circuito de la placa

7.3.7. Código básico

explicacion de lo minimo que deberia tener para ser parte de la cadena de comunicacion y sensado de los distintos perifericos

7.3.8. Posibles extensiones

uso de componentes como resistencias variables para regular la alimentacion de los sensores de piso y resistencias pull-up, pasar a montaje superficial los componentes, hacerla mas chica

7.4. Placa controladora de servo motores

funcion de una placa controladora de servos, porque no fue armada, para que se penso, alguna otra opcion de conexion, pines libres

7.4.1. Características principales

principio de funcionamiento, como logra generar varios pwm por software, como logra ser parte de la cadena

7.4.2. Módulo de comunicación

se explico en el modulo generico, se agregan los comandos especificos y se puede explicar como se obtiene la informacion para dar las respuestas

7.4.3. Alimentación de la placa

se explico en el modulo generico, con modificaciones que permiten que circule una mayor cantidad de corriente para alimentar a los servos.

7.4.4. Configuración

configuracion de la placa, comunicacion, header de programacion

7.4.5. Esquemático

esquematicos de la placa

7.4.6. Circuito

circuito de la placa

7.4.7. Código básico

explicacion de lo minimo que deberia tener para ser parte de la cadena de comunicacion y control de los servos

7.4.8. Posibles extensiones

uso de componentes como resistencias variables para regular la alimentacion de los sensores de piso y resistencias pull-up, pasar a montaje superficial los componentes, hacerla mas chica

8. Armado del prototipo

8.1. Diseño

8.2. Características

con las ruedas de 10cm y teniendo en cuenta que el motor gira a unas 300 cuentas/segundo (usando un solo sensor en el encoder) llegamos a 50 cm/s de velocidad

8.3. Desarme

8.4. Costo y proveedores

A. Primer apéndice Hardware

protocolo de comunicacion.
conexionado y configuracion de la comunicacion.
circuitos de las placas.
codigo fuente de las controladoras.
costo del prototipo.

B. Segundo apéndice Hardware