

**ESCUELA DE DOCTORADO
Y ESTUDIOS DE POSGRADO**

Trabajo de Fin de Máster

**Desarrollo de la adaptación para multicámara
de la Nvidia Jetson TX2**

Titulación: Máster en Ingeniería Industrial

Estudiante: Iván Rodríguez Méndez

Tutor: Fernando Luis Rosa González

25 de junio de 2018

**IMPRESO DE AUTORIZACIÓN DEL
TRABAJO DE FIN DE MÁSTER POR EL
TUTOR**

Curso 2017/2018

El Dr. D. **Fernando Luis Rosa González**, con D.N.I. 43611314-W, como tutor del estudiante D. **Iván Rodríguez Méndez** en el Trabajo de Fin de Máster titulado

Desarrollo de la adaptación para multicámara de la Nvidia Jetson TX2,

da su autorización, acreditada por la firma electrónica de este documento, para la presentación y defensa de dicho proyecto, a la vez que confirma que el estudiante ha cumplido con los objetivos generales y particulares que lleva consigo la realización del mismo. Quiero hacer mención especial del aprovechamiento del estudiante que ha sobresalido de la media en la realización de este trabajo.

La Laguna, a 25 de junio de 2018

Agradecimientos

La realización de este Trabajo de Fin de Máster no habría sido posible sin la ayuda incondicional de mi tutor Fernando Rosa. No puedo estar más agradecido por la confianza que ha depositado en mí para el desarrollo de semejante proyecto. Gracias por enseñarme a pensar de una forma totalmente diferente a la que estaba acostumbrado. Por ser siempre inconformista forzándome así a mejorar en cada paso que he dado. Gracias por enseñarme que el verdadero maestro no es aquel que enseña un camino ideal, sino el que muestra a su alumno las muchas vías de acceso hasta el camino que él tendrá que recorrer para encontrarse con su destino. Por otro lado, también me gustaría agradecer a Delfín su gran labor y ayuda en los momentos en que tocaba mancharse las manos. Gracias por todo lo que me has enseñado y por tener siempre tiempo para atender todas mis peticiones.

Doy las gracias también a mis padres por haber estado ahí, por seguir recor-dándome que la humildad y el trabajo duro son las piezas fundamentales del éxito. Gracias por el esfuerzo para que yo llegara hasta aquí. Doy gracias a mi padre por prestarme la mayoría de herramientas con las que pude fabricar el prototipo de este proyecto. Gracias por los consejos que me has dado, de ti he aprendido que con esfuerzo todo se consigue. Gracias a mi madre por estar ahí, por preguntarme cada día que pasaba en el laboratorio si había comido o por cómo me estaban saliendo las cosas ese día. Gracias por predicar con el ejemplo, por no tener nunca una mala palabra, una mala contestación y estar siempre tan radiante y feliz. Gracias también a mi hermana por su apoyo, le deseo lo mejor en la etapa universitaria que recién está comenzando. Como siempre nos han dicho, con esfuerzo vas a llegar allá donde te propongas. Muchísimas gracias también a Noemí, mi pareja, que ha estado ahí día tras día siendo mi apoyo fundamental. Gracias por soportar que mi único tema de conversación fuera este proyecto, por darme consejo, por siempre tener una sonrisa y preocuparte por alegrarme los días malos. Gracias por entenderme y sobre todo por aguantar mi carácter en los momentos de frustración. Gracias por confiar en mi ciegamente hasta en los momentos en que yo mismo pierdo la esperanza.

No quiero olvidarme de agradecer el apoyo a mis compañeros del *Orotava HackerSpace*. Muchas gracias Tana, Ricardo y Toño por echarme un cable siempre que me hizo falta.

Por último, gracias a ti abuela, gracias por haber sido la persona que me ha enseñado a nunca darme por vencido aunque todo esté en contra.

Muchísimas gracias a todos.

**ESCUELA DE DOCTORADO
Y ESTUDIOS DE POSGRADO**

Trabajo de Fin de Máster

**Desarrollo de la adaptación para multicámara
de la Nvidia Jetson TX2**

**TOMO I
Memoria**

Titulación: Máster en Ingeniería Industrial

Estudiante: Iván Rodríguez Méndez

Tutor: Fernando Luis Rosa González

25 de junio de 2018

Índice general

| | |
|---|-----------|
| I Memoria | 7 |
| Resumen | 21 |
| Abstract | 23 |
| 1. Introducción | 25 |
| 1.1. Marco general | 25 |
| 1.2. Objetivos | 26 |
| 1.3. Cronología del desarrollo | 27 |
| 1.4. Distribución de esta memoria | 28 |
| 1 Introduction | 31 |
| 2. Material y método | 35 |
| 2.1. Material | 35 |
| 2.1.1. Distribuciones Linux | 35 |
| 2.1.2. Git | 36 |
| 2.1.3. <i>KiCad EDA</i> | 36 |
| 2.1.4. <i>FreeRoute</i> | 39 |
| 2.1.5. <i>Freecad</i> | 39 |
| 2.1.6. <i>Inkscape</i> | 39 |
| 2.1.7. <i>FlatCAM</i> | 40 |
| 2.1.8. <i>CAMotics</i> | 40 |
| 2.1.9. <i>Slic3r</i> | 40 |
| 2.1.10. <i>Python</i> | 41 |
| 2.1.11. <i>OpenCV</i> | 41 |
| 2.1.12. <i>LATeX</i> y Kile | 41 |
| 2.1.13. Máquinas CNC (control numérico computerizado) | 42 |
| 2.1.14. Horno de soldadura | 43 |
| 2.1.15. Impresoras 3D | 43 |
| 2.1.16. Cortadora láser | 43 |

| | |
|---|-----------|
| 2.1.17. Lupa electrónica | 45 |
| 2.1.18. Nvidia Jetson TX2 | 45 |
| 2.2. Método | 47 |
| 3. Diseño e integración del sistema | 49 |
| 3.1. Descripción general del sistema | 49 |
| 3.2. Descripción de elementos del sistema | 50 |
| 3.2.1. Array de cámaras | 50 |
| 3.2.2. Nvidia Jetson TX2 | 52 |
| 3.2.3. Estructura mecánica de soporte | 56 |
| 4. Diseño de la electrónica | 59 |
| 4.1. Módulos principales a diseñar | 59 |
| 4.1.1. Módulo de cámara | 61 |
| 4.1.2. Módulo puente | 61 |
| 4.1.3. Módulo de depuración | 62 |
| 4.2. Restricciones de diseño | 62 |
| 4.3. Direccionamiento i^2c | 63 |
| 4.4. Metodología de diseño | 64 |
| 4.4.1. Creación de componentes | 64 |
| 4.4.2. Creación de huellas | 64 |
| 4.4.3. Diseño esquemático | 65 |
| 4.4.4. Diseño de circuito impreso | 65 |
| 4.5. Diseño preliminar | 66 |
| 4.5.1. Consideraciones de diseño previas | 66 |
| 4.5.2. Módulos de cámara diseñados | 68 |
| 4.5.3. Módulos puente diseñados | 70 |
| 4.5.4. Esquemas eléctricos diseñados | 73 |
| 4.5.5. Diseño de circuitos impresos | 74 |
| 4.5.6. Modelos 3D de las placas diseñadas | 78 |
| 4.6. Viabilidad de diseños preliminares | 79 |
| 4.7. Módulos a implementar en el desarrollo | 80 |
| 5. Fabricación de PCBs | 83 |
| 5.1. Características físicas de las placas | 83 |
| 5.2. Metodología de fabricación de PCBs | 84 |
| 5.3. Fabricación del módulo de depuración | 88 |
| 5.3.1. Proceso de fabricación de la PCB | 88 |
| 5.3.2. Soldadura de conectores | 92 |
| 5.3.3. Aplicación de laca a la PCB | 93 |
| 5.3.4. Resultado final | 94 |

| | |
|--|------------|
| ÍNDICE GENERAL | 11 |
| 6. Diseño y fabricación mecánica | 97 |
| 6.1. Diseño de la estructura mecánica | 97 |
| 6.1.1. Elementos principales a diseñar | 97 |
| 6.1.2. Metodología de diseño | 98 |
| 6.1.3. Elementos diseñados | 99 |
| 6.2. Fabricación de elementos mecánicos | 102 |
| 6.2.1. Características mecánicas de las piezas | 102 |
| 6.2.2. Metodología de fabricación de piezas | 103 |
| 6.2.3. Fabricación del calibrador de cámaras | 104 |
| 6.2.4. Fabricación de la estructura mecánica de soporte | 104 |
| 6.3. Integración final | 107 |
| 7. Configuración del Kernel y el <i>Device tree</i> | 109 |
| 7.1. Preparación de elementos | 109 |
| 7.1.1. <i>Hardware</i> necesario | 109 |
| 7.1.2. <i>Software</i> necesario | 110 |
| 7.2. Desarrollo de software para la detección del sensor de cámara | 110 |
| 7.2.1. Preparación de ficheros fuente del <i>Device tree</i> | 111 |
| 7.2.2. Modificación de ficheros fuente del <i>device tree</i> | 112 |
| 7.2.3. Modificación de drivers para módulos de cámara y compilación del kernel | 117 |
| 7.3. Detección de dispositivos usando el conector J20 | 118 |
| 8. Adquisición y procesado de imágenes | 119 |
| 8.1. Adquisición de imágenes | 119 |
| 8.1.1. <i>Gstreamer-1.0</i> | 119 |
| 8.1.2. <i>V4l2</i> | 120 |
| 8.2. Tratamiento de los datos <i>raw</i> | 121 |
| 8.3. Calibración de cámaras | 121 |
| 8.3.1. Calibración física de las cámaras | 122 |
| 8.3.2. Calibración por software | 123 |
| 8.4. Generación del <i>Light Field</i> | 125 |
| 8.5. Estimación del mapa de disparidad | 126 |
| 9. Resultados de la estimación de disparidad | 129 |
| 9.1. Resultados para la primera escena | 129 |
| 9.2. Resultados para la segunda escena | 130 |
| 10. Conclusiones | 135 |
| 10 Conclusions | 137 |

| | |
|---|------------|
| II Pliego de condiciones y presupuesto | 143 |
| 11. Pliego de condiciones | 145 |
| 12. Presupuesto del proyecto | 147 |
| | |
| III Anexos | 149 |
| A. Cronología del proyecto | 151 |
| A.1. Diagrama de Gantt | 152 |
| A.2. Cuadro de temporalización | 153 |
| B. Datasheets de los Integrados y componentes utilizados | 155 |
| B.1. Conector Molex FPC 30 pin | 156 |
| B.2. Ceramic wire wound Inductors | 157 |
| B.3. EEPROM AT24C01D | 158 |
| B.4. Filtro EMI EClamp2398P | 159 |
| B.5. Conector sensor OV5680 - AXT630124 | 160 |
| B.6. Chip Inductors | 161 |
| B.7. Conector QSH-060-01-H-D-A | 162 |
| B.8. TCA9548A | 163 |
| B.9. Cable Molex FPC 30 pin | 164 |
| B.10. C1206C120J5GACTU | 165 |
| B.11. C1206C104J5RACTU | 166 |
| B.12. Multilayer Ceramic Chip Capacitors | 167 |
| B.13. Think film chip Resistors | 168 |
| B.14. 1SMA5917BT3G | 169 |
| B.15. Conector de pines | 170 |
| C. Manuales de referencia de la NVIDIA JETSON TX2 | 171 |
| C.1. Developer Kit Carrier Board Specification | 172 |
| C.2. OEM Product Design Guide | 173 |
| C.3. Module DataSheet | 174 |
| C.4. OEM Product Design Guide | 175 |
| C.5. Camera Module Hardware Design Guide | 176 |
| C.6. MIPI DSI/CSI Design and Tuning | 177 |
| C.7. Manual de referencia del conector Auvidea J20 | 178 |
| D. Hojas de características NVIDIA JETSON TX2 | 179 |
| D.1. Especificaciones eléctricas | 183 |

| | |
|---|------------|
| ÍNDICE GENERAL | 13 |
| E. Hojas de características de sensores CMOS | 185 |
| E.1. OmniVision OV5680 | 186 |
| E.2. OmniVision OV5693 | 187 |
| E.3. OmniVision OV5647 | 188 |
| E.4. OmniVision OV5640 | 189 |
| E.5. Sony IMX219 | 190 |
| F. Hojas de esquema del diseño electrónico | 191 |
| G. Planos de diseño de placas de circuito impreso | 197 |
| H. Modelos 3D de los módulos desarrollados | 245 |
| I. Características de los componentes | 257 |
| I.1. Componentes para el módulo del sensor de cámara OV5680 | 257 |
| I.2. Componentes para el módulo puente sensor OV5680 | 258 |
| I.3. Componentes módulo puente para el sensor IMX219/OV5647 | 258 |
| I.4. Componentes módulo puente con conexión DUPONT | 259 |
| I.5. Componentes módulo puente con conexión DUPONT sin EEPROM | 259 |
| I.6. Componentes módulo de depuración de líneas | 260 |
| J. Planos de diseño del soporte móvil de cámaras | 261 |

Índice de figuras

| | |
|---|----|
| 2.1. CNC–4030Z | 42 |
| 2.2. CNC–Shapeoko–3–XL | 43 |
| 2.3. Horno de soldadura T962 | 44 |
| 2.4. Impresora 3D Prusa i3 Hephestos | 44 |
| 2.5. Cortadora láser del Orotava Hackerspace | 45 |
| 2.6. Lupa electrónica del laboratorio de Comunicaciones y Teledetección | 46 |
| 2.7. Kit de desarrollo NVIDIA Jetson TX2 | 46 |
| 2.8. Características de los módulos NVIDIA Jetson | 47 |
| 3.1. Diagrama de bloques de elementos del sistema | 50 |
| 3.2. Diagrama de bloques del flujo de datos del <i>array</i> de cámaras | 51 |
| 3.3. Giros para el cambio de orientación de los sensores de cámara | 52 |
| 3.4. Diagrama de señales del módulo puente | 53 |
| 3.5. Diagrama de traducción de direcciones en módulo puente | 54 |
| 3.6. Diagrama de relación entre las cámaras y los elementos software del <i>device tree</i> | 55 |
| 3.7. Adquisición y tratamiento de imágenes | 56 |
| 4.1. Diagrama de diseño del módulo puente | 60 |
| 4.2. Diagrama de diseño del módulo de cámara | 61 |
| 4.3. Restricciones geométricas del módulo puente | 63 |
| 4.4. Circuitos para la adecuación de las alimentaciones | 67 |
| 4.5. Configuración de EEPROM para el módulos de cámara | 67 |
| 4.6. Configuración de EEPROM para el módulos puente | 68 |
| 4.7. Filtro para interferencias electromagnéticas | 68 |
| 4.8. Sensor y módulos de cámara utilizados | 69 |
| 4.9. Distribución de los componentes del módulo de cámara para OV5680 | 75 |
| 4.10. Distribución de los componentes del módulo puente para sensor OV5680 | 76 |
| 4.11. Trazado de pistas en el diseño PCB del módulo de cámara para el sensor OV5680 | 76 |

| | |
|--|-----|
| 4.12. Trazado de pistas en el diseño PCB del módulo puente para sensor OV5680 | 77 |
| 4.13. Visualización 3D del módulo de cámara para el sensor OV5680 | 78 |
| 4.14. Visualización 3D del módulo puente para el sensor OV5680 | 79 |
| 4.15. Visualización 3D del módulo de depuración | 80 |
| 4.16. Módulo J20 | 81 |
| 5.1. Visualización 3D de ambas caras del módulo de depuración. | 87 |
| 5.2. Placas de cobre de doble cara | 88 |
| 5.3. Proceso de transferencia de fotolito por plastificadora de rodillos | 89 |
| 5.4. Resultado de la transferencia del toner sobre la placa de cobre. | 89 |
| 5.5. Ataque químico a la placa de cobre | 90 |
| 5.6. Resultado del ataque químico la placa de cobre. | 90 |
| 5.7. Resultado de la limpieza con acetona de la PCB. | 91 |
| 5.8. Corte de pistas con bisturí | 91 |
| 5.9. Dremel para realizar los agujeros de los conectores | 92 |
| 5.10. Proceso de soldadura con horno. | 92 |
| 5.11. Resultado del proceso de soldadura. | 93 |
| 5.12. Protecciones para aplicar laca a la PCB. | 93 |
| 5.13. Resultado final de la fabricación del módulo de depuración. | 94 |
| 5.14. Módulo de depuración integrado en Nvidia Jetson TX2 | 95 |
| 6.1. Diseño del módulo de calibración de cámaras | 99 |
| 6.2. Modelo 3D de fijación de tornillos en el módulo de calibración | 100 |
| 6.3. Modelo 3D del calibrador para 6 cámaras | 100 |
| 6.4. Modelo 3D de la estructura de soporte del sistema. | 101 |
| 6.5. Módulo de calibración de cámaras impreso | 104 |
| 6.6. Piezas de estructura de soporte impresas | 105 |
| 6.7. Estructura básica de soporte ensamblada | 106 |
| 6.8. Mesa de soporte de NJTX2 fabricada | 106 |
| 6.9. Estructura de soporte del sistema fabricada. | 108 |
| 7.1. Diagrama de integración de los módulos que componen el DTS [28] | 113 |
| 7.2. Montaje para test de líneas con el módulo de depuración | 114 |
| 8.1. Streaming de vídeo en tiempo real de 6 cámaras con <i>Gstreamer-1.0</i> | 120 |
| 8.2. Captura realizada con <i>v4l2</i> desde una de las cámaras del array (Imagen Posprocesada) | 120 |
| 8.3. Comparativa entre datos raw y datos tratados. | 121 |
| 8.4. Montaje para ajustar el ángulo de apuntado de las cámaras | 122 |
| 8.5. Ajuste de apuntado de cámaras. | 123 |
| 8.6. Patrón para el cálculo de parámetros de cámaras | 123 |

| | |
|---|-----|
| 8.7. Ejemplo de capturas de calibración | 124 |
| 8.8. Detección de corners | 125 |
| 8.9. <i>Light Field</i> generado | 125 |
| 8.10. Proceso de estimación del mapa de distancias | 126 |
| 8.11. Resultados de la estimación de distancias | 127 |
| 9.1. Montaje e iluminación de una escena | 130 |
| 9.2. Resultados de la estimación de la escena 1. | 131 |
| 9.3. Resultados de la estimación de la escena 2. | 132 |
| H.1. Visualización 3D del módulo de cámara para el sensor OV5680 | 246 |
| H.2. Visualización 3D del módulo de cámara para el sensor OV5680 | 247 |
| H.3. Visualización 3D del módulo puente para el sensor OV5680 | 248 |
| H.4. Visualización 3D del módulo puente para el sensor OV5680 | 249 |
| H.5. Visualización 3D del módulo puente para el sensor IMX219/OV5647 | 250 |
| H.6. Visualización 3D del módulo puente para el sensor IMX219/OV5647 | 251 |
| H.7. Visualización 3D del módulo puente para el sensor IMX219/OV5647 con conexión dupont | 252 |
| H.8. Visualización 3D del módulo puente para el sensor IMX219/OV5647 con conexión dupont | 253 |
| H.9. Visualización 3D del módulo de depuración | 254 |
| H.10. Visualización 3D del módulo de depuración | 255 |
| H.11. Visualización 3D módulo de pruebas para CNC | 256 |

Índice de tablas

| | | |
|-------|--|-----|
| 4.1. | Alimentaciones disponibles en el conector J22 | 62 |
| 4.2. | Direccionamiento i^2c de integrados para la Nvidia Jetson TX2 | 64 |
| 4.3. | Componentes del módulo de cámara OV5680 | 70 |
| 4.4. | Componentes del módulo puente para el sensor OV5680 | 71 |
| 4.5. | Componentes del módulo puente para el sensor OV5647 | 72 |
| 4.6. | Componentes del módulo puente para el sensor OV5647/IMX219 con conexión DUPONT | 72 |
| 4.7. | Componentes del módulo de depuración de líneas. | 73 |
| 4.8. | Valores mínimos para el trazado de pistas | 75 |
| 7.1. | Direcciones i2c y alias en la Nvidia Jetson TX2 | 115 |
| 7.2. | Configuración de direcciones i2c en los ficheros DTS para la conexión de 6 cámaras | 116 |
| 7.3. | Detección de dispositivos i^2c en el módulo puente J20 de Auvida [17] | 118 |
| 12.1. | Descomposición del presupuesto del proyecto | 147 |
| 12.2. | Precio detallado de los componentes mecánicos | 148 |
| 12.3. | Precio detallado de los componentes electrónicos | 148 |
| I.1. | Características de los componentes del módulo de cámara OV5680 | 258 |
| I.2. | Características de los componentes del módulo puente para el sensor OV5680 | 258 |
| I.3. | Características de los componentes del módulo puente para el sensor OV5647 | 259 |
| I.4. | Características de los componentes del módulo puente para el sensor OV5647/IMX219 con conexión DUPONT | 259 |
| I.5. | Características de los componentes del módulo puente para el sensor OV5647/IMX219 con conexión DUPONT sin EEPROM | 260 |
| I.6. | Características de los componentes del módulo de depuración de líneas. | 260 |

Resumen

En el presente Trabajo de Fin de Máster se ha desarrollado el hardware y software necesario para la correcta adaptación de un array de cámaras utilizando la placa de desarrollo NVIDIA Jetson TX2. Se ha llevado a cabo el diseño electrónico de los diferentes elementos así como el diseño estructural para el soporte mecánico del array y de algunas piezas para su calibración física. Se ha realizado la fabricación de los diferentes elementos, tanto mecánicos como electrónicos, y se ha desarrollado el software necesario tanto en la configuración del *kernel* como la del *device tree* para el correcto funcionamiento de las cámaras. Por último, se ha propuesto como ejemplo de uso la calibración por software del array de cámaras y un algoritmo para la estimación de mapas de disparidad.

Abstract

In the present Master's Degree Final Project the necessary hardware and software has been developed for the correct adaptation of an array of cameras using the NVIDIA Jetson TX2 development board. The design has been carried out of the different elements as well as the structural design for the the mechanical support of the array and some parts for physical calibration. The fabrication of the different mechanical and electronic elements, and the necessary software has been developed in the configuration of the kernel and the Device tree for the correct operation of the cameras. Finally, the software calibration of the array of cameras and an algorithm for the estimation of disparity maps have been proposed.

Capítulo 1

Introducción

1.1. Marco general

En general, se acepta como uno de los principios económicos de la industria tecnológica moderna, que los productos que se venden en grandes cantidades son más baratos que los que se venden en cantidades más bajas con independencia de la complejidad tecnológica. En el caso de la informática, esta relación ha cambiado la forma en que los ingenieros piensan sobre la construcción de sistemas de alta gama. A la hora de construir un procesador de gama alta personalizado, se tiene como opción más rentable la utilización de una gran cantidad de procesadores de productos más básicos. En el caso de los productos relacionados con las imágenes digitales, esto no es diferente. Como la popularidad de las cámaras digitales crece, el rendimiento de los lectores de imágenes de gama baja ha crecido, mientras que el costo de las cámaras de gama alta permanece relativamente constante. Además, se ha demostrado que múltiples imágenes de una escena estática se pueden usar para mejorar el rendimiento de estas cámaras [34]. Algunos ejemplos de esta mejora son crear imágenes con mayor resolución [30] o con mayor rango dinámico [15]. Otro uso común para múltiples vistas, es la interpolación de las imágenes para crear una cámara virtual que se mueva suavemente en una escena estática o dinámica [23].

En este sentido, la tecnología actual de sistemas de procesamiento de datos en paralelo basados en GPU's, como las lideradas por Nvidia, permite diseñar sistemas que pueden realizar la tarea de extracción de información geométrica de las escenas en tiempos reducidos. En este proyecto se cuenta con una plataforma Jetson TX2 que permite la gestión masiva de datos paralelos por medio de sus GPU's [20]. En esta plataforma el fabricante ha dispuesto un conector de cámara mediante los protocolos MIPI CSI-2 [25], protocolos de comunicaciones seriales utilizados ampliamente en los sectores de automoción y periféricos móviles, cuyo

objetivo es permitir la comunicación punto a punto tanto de la imagen como del vídeo, entre las cámaras y los sistemas de gestión de datos. La toma de imágenes desde varias cámaras puede realizarse de modo que se estructuren *Light Fields* o con otras geometrías multivista que permitan inferir de modo automático la estructura geométrica de la escena [13].

En este trabajo, se utiliza una **Nvidia Jetson TX2** para diseñar y poner en funcionamiento un sistema capaz de adquirir imágenes desde sensores CMOS de bajo coste que permitan su uso en la extracción de información geométrica de la escena.

1.2. Objetivos

El objetivo principal de este proyecto es el desarrollo de la adaptación para multicámera en la plataforma de desarrollo **Nvidia Jetson TX2** además del diseño mecánico para su soporte y calibración. Para la consecución de este objetivo es necesario la realización de una serie de sub–objetivos que se relacionan a continuación:

- Diseñar una placa electrónica capaz de adquirir los datos procedentes de un sensor CMOS.
- Diseñar una interfaz electrónica para la conexión de múltiples cámaras a la **Nvidia Jetson TX2**.
- Fabricar prototipos de placas de circuito impreso que permitan una mejor comprensión de la plataforma utilizada.
- Desarrollar la integración hardware-software para la **Nvidia Jetson TX2**.
- Desarrollar software de adquisición de imágenes con el uso de *v4l2-src* y *Gstreamer-1.0* .
- Diseñar una estructura mecánica capaz de sustentar el sistema de adquisición.
- Diseñar un sistema para calibrar el ángulo de apuntado de las cámaras.
- Fabricar por medio de técnicas de prototipado rápido prototipos funcionales de la mecánica necesaria.
- Integrar todos los elementos diseñados.
- Desarrollar la calibración de la matriz de cámaras por medio de *software*.

- Comprobar el funcionamiento de la matriz de cámaras desarrollada con un algoritmo de estimación de disparidad.

El proyecto abarca el diseño, fabricación y puesta en funcionamiento de un sistema multicámara para la Nvidia Jetson TX2. Las particularidades que se refieren a la adaptación multicámara consisten en analizar el funcionamiento de las líneas CSI para el tratamiento de los datos procedentes de los sensores de cada cámara. Además se diseñará toda la electrónica necesaria para ello y se desarrollará el *software* para la correcta comunicación con la placa de desarrollo utilizada. Se finalizará con un caso de uso consistente en la adaptación de un algoritmo sencillo sin etapa de regularización para la estimación de un mapa de disparidad disperso.

1.3. Cronología del desarrollo

Este proyecto se ha desarrollado de forma secuencial realizando las tareas en 4 paquetes de trabajo siguiendo el orden siguiente:

1. Desarrollo electrónico.
2. Desarrollo mecánico.
3. Desarrollo de software.
4. Redacción del proyecto.

El proyecto se inició acometiendo el desarrollo electrónico, donde se estudió el estado del arte con respecto a la integración de sensores CMOS en placas de desarrollo Nvidia Jetson, esta tarea tuvo una duración de una semana. Seguidamente se recopiló información sobre la placa de desarrollo que usaríamos para el desarrollo del proyecto, empleando una semana para ello. Posteriormente, se realizaron los diseños electrónicos para la integración de los dispositivos en la Nvidia Jetson (duración de 3 semanas) y se realizaron los pedidos de los componentes necesarios (duración de dos días). Finalmente, se estudiaron y se realizaron distintas pruebas sobre las tecnologías de fabricación de PCBs con máquinas de control numérico y con químicos, con una duración de 3 semanas. Simultáneamente también se realizaron test sobre las placas fabricadas con la misma duración temporal que la fabricación de las mismas.

Mientras se esperaba la llegada de los componentes electrónicos para la fabricación de las placas de circuito impreso, se comenzó con el desarrollo mecánico del proyecto. La primera tarea de este paquete de trabajo fue el estudio del estado del arte en cuanto a la construcción de *arrays* para el posicionamiento de cámaras. Esta tarea tuvo una duración de 2 días aproximadamente. Posteriormente, se

realizaron los diseños de las piezas mecánicas con una duración de 2 semanas y se fabricaron en una semana. Finalmente, se ensambló la estructura, se incorporaron las cámaras y la Nvidia Jetson. Todo ello con una duración aproximada de 1 semana.

Una vez finalizados los paquetes de trabajo del desarrollo electrónico y mecánico, se procedió al desarrollo del software. Se comenzó, al igual que en el resto de paquetes de trabajo, realizando un estudio del estado del arte para analizar la existencia de otros desarrollos similares. Esta tarea tuvo una duración de 1 semana. Seguidamente, se comenzó el desarrollo del software para el reconocimiento del hardware instalado en la Nvidia Jetson. Esto tuvo una duración de 4 semanas. A continuación, se desarrolló el software para la obtención de imágenes de las cámaras durante 2 semanas. En la siguiente semana se trabajó en el software de calibración de las cámaras y en la toma de imágenes para dicha calibración. En la última semana, se tomaron fotografías en un escenario medido y se desarrolló el software para la obtención de *light fields* además de un algoritmo sencillo de estimación de mapas de disparidad.

Finalmente, una vez finalizados los 3 grandes paquetes de trabajo, se comenzó con la redacción del proyecto y toda la documentación, con una duración de 2 semanas aproximadamente.

Para una mejor comprensión de la cronología de desarrolló de este proyecto, se puede consultar el Anexo A, donde se muestra el diagrama de Gantt en la Figura A.1 y la temporalización en la Figura A.2.

1.4. Distribución de esta memoria

Esta memoria se distribuye en tres partes, la *Memoria*, el *Pliego de Condiciones y Presupuesto* y los *Anexos*.

En la parte de la *Memoria*, además de establecer los objetivos del proyecto y la cronología de las fases llevadas a cabo para su elaboración en este capítulo, se describe el material y los métodos utilizados para el desarrollo del proyecto en el capítulo 2. En el capítulo 3 se presenta una descripción general de los elementos que componen el sistema. La descripción detallada de cada una de las etapas de diseño electrónico se relatan en el capítulo 4 y los pasos seguidos para la fabricación se exponen en el capítulo 5. El diseño y fabricación de la mecánica se expone en el capítulo 6. La configuración del *kernel* y el *device tree* se trata en el capítulo 7. En el capítulo 8 se explican los métodos para la adquisición y el procesamiento de las imágenes. Por último, en el capítulo 9 se muestran los resultados obtenidos para la estimación de disparidad y por otro lado en el capítulo 10 se muestran las conclusiones y líneas abiertas a modo de finalización de la memoria.

En la parte del *Pliego de Condiciones y Presupuesto* se exponen los condicionantes que tenía el proyecto para su elaboración antes de comenzarlo y también el presupuesto resultante del desarrollo del proyecto así como el coste de fabricación del prototipo.

La última parte de *Anexos* contiene los listados que por su tedio y longitud se han extraído del texto de la memoria para que sirvan de referencia al lector interesado.

Introduction

General framework

It is generally accepted as one of the economic principles of the modern technology industry that products sold in large quantities are cheaper than those sold in smaller quantities regardless of technological complexity. In the case of computing, this relationship has changed the way engineers think about building high-end systems. When building a custom high-end processor, the most cost-effective option is to use a large number of more basic product processors. In the case of products related to digital imaging, this is no different. As the popularity of digital cameras grows, the performance of low-end imagers has grown, while the cost of high-end cameras remains relatively constant. In addition, researchers have shown that multiple images of a static scene can be used to improve the performance of the set of these cameras [34]. Some examples of the above include creating images with higher resolution [30] or dynamic range [15]. Another common use for multiple views is interpolating images to create a virtual camera that moves smoothly in a static or dynamic scene [23].

The current technology of parallel data processing systems based on GPU's such as those led by Nvidia allow the design of systems that can perform the task of geometric measurement of the scenes in fairly short times. For this purpose, it has a Jetson TX2 platform that allows the massive management of parallel data through its GPUs [20]. On the same platform, the manufacturer has provided a camera connector using the MIPI CSI-2 [25] protocols. This protocol is the serial communications protocol widely used in the automotive and mobile peripherals industries and aims to enable point-to-point image and video communication between cameras and data management systems. Imaging from multiple cameras can be done in a way that structures Light Fields or with other multiview geometries to automatically infer the geometric structure of the scene [13].

In this work, a textbf{Nvidia Jetson TX2} is used to design and implement a system capable of acquiring images from low-cost CMOS sensors that can be used to extract geometric information from the scene.

Objectives

The main objective of this project is the development of the multi-camera adaptation in the development platform **Nvidia Jetson TX2** as well as the mechanical design for its support and calibration. In order to achieve this objective, a series of sub-objectives are necessary, which are listed below:

- Design an electronic board capable of acquiring data from a CMOS sensor.
- Design an electronic interface for the connection of multiple cameras to the Nvidia Jetson TX2.
- Fabricate prototypes of printed circuit boards for a better understanding of the platform used.
- Develop the hardware-software integration for the **Nvidia Jetson TX2**.
- Develop image acquisition software using textv4l2-src and textGstreamer-1.0.
- Design a mechanical structure capable of supporting the acquisition system.
- Design a system to calibrate the angle of aim of the cameras.
- Fabricate functional prototypes of the required mechanics using rapid prototyping techniques.
- Integrate all the designed elements.
- Develop the calibration of the camera array by software.
- Check the functioning of the camera array developed with a camera estimation algorithm of disparity.

The project includes the design, fabrication and implementation of a multi-camera system for the Nvidia Jetson TX2. The special features of the multi-camera adaptation are the analysis of the operation of the lines CSI for the processing of data from the sensors of each camera. In addition, all the necessary electronics will be designed and the software for correct communication will be developed with the development board used. It will end with a case of use consisting of the adaptation of an algorithm simple without regularization stage for the estimation of a dispersed disparity map.

Chronology of development

This project has been developed in a sequential way performing the tasks in 4 work packages following the following order:

1. Electronics development.
2. Mechanical development.
3. Software development.
4. Project writing.

The project began with the electronic development, where the state of the art was studied with reference to the integration of CMOS sensors in Nvidia Jetson development boards, this task lasted one week. Afterwards, information was gathered about the development board that we would use for the development of the project, using one week for this purpose. Subsequently, the electronic designs for the integration of the devices into the Nvidia Jetson were made (duration of 3 weeks) and the necessary components were ordered (duration of 2 days). Finally, different tests on PCB fabrication technologies with numerical control machines and chemicals were studied and carried out, with a duration of 3 weeks. Simultaneously, testing was also carried out on the boards made with the same time duration as their production.

While awaiting the arrival of the electronic components for the production of the printed circuit boards, the mechanical development of the project began. The first task of this work package was to study the state of the art in terms of the construction of arrays for camera positioning. This task lasted approximately 2 days. The mechanical parts were then designed for a duration of 2 weeks and fabricated in one week. Finally, the structure was assembled, the cameras and the Nvidia Jetson were incorporated. All this with an approximate duration of 1 week.

Once the electronic and mechanical development work packages were completed, the software was developed. As with the rest of the work packages, a study of the state of the art was begun to analyse the existence of other similar developments. This task lasted 1 week. Subsequently, the development of the software for the recognition of the hardware installed on the Nvidia Jetson was started. This lasted 4 weeks. The software for obtaining images from the cameras was then developed for 2 weeks. In the following week, work was done on the camera calibration software and on taking images for this calibration. In the last week, photographs were taken in a measured scenario and software was developed to obtain the textitlight fields as well as a simple estimation algorithm of disparity maps.

Finally, once the 3 large work packages had been completed, the project all the documentation were prepared , with a duration of approximately 2 weeks.

For a better understanding of the chronology of development of this project, consult Annex A, where the Gantt chart is shown in Figure A.1 and the timing in Figure A.2.

Distribution of this report

This report is divided into three parts, the Report, the Specification and Budget and the Annexes.

In the section of the Report, in addition to setting out the objectives of the project and the chronology of the phases carried out for its development in this chapter, the material and methods used for the development of the project are described in the chapter 2. An overview of the elements of the system is given in chapter 3. The detailed description of each of the stages of electronic design is described in chapter 4 and the steps followed for manufacturing are described in chapter 5. The design and manufacture of the mechanics is described in chapter 6. The configuration of the texttitkernel and the texttitdevice tree is covered in chapter 7. Chapter 8 explains the methods for image acquisition and processing. Finally, in chapter 9 the results obtained for the disparity estimation are shown and in chapter 10 the conclusions and open lines are shown as the end of the report.

In the part of the Specifications and Budget, the conditions that the project had for its preparation before it began and also the budget resulting from the development of the project as well as the cost of manufacturing the prototype are set out.

The last part of the Annexes contains the lists that, due to their tedium and length, have been extracted from the text of the report to serve as a reference for the interested reader.

Capítulo 2

Material y método

2.1. Material

Tanto el *software* como parte del *hardware* empleados son de tipo libre u *open source*, es decir, sus especificaciones, códigos y diagramas esquemáticos son de acceso público, ya sea bajo algún tipo de pago o de forma gratuita. Así, el término “libre” hace referencia a que, por elección manifiesta de su autor, este tipo de material puede ser copiado, estudiado, modificado, utilizado libremente para cualquier fin y redistribuido con o sin cambios y/o mejoras.

Para la realización del proyecto se necesita trabajar sobre un sistema operativo GNU/Linux con una distribución igual al resto del proyecto para poder intercambiar información con el resto del equipo de trabajo, o al menos las versiones de los diferentes paquetes de software a utilizar deberán ser las mismas para evitar problemas. Para llevar control de las versiones realizadas, el equipo de trabajo utiliza *Git*. Para el diseño electrónico se utiliza la suite *KiCad*, para el diseño meccanico se usa *FreeCAD*, para el desarrollo de software *Python* y para la elaboración de la documentación se utiliza *LATEX*. Las diferentes aplicaciones físicas y virtuales libres de las que se ha hecho uso durante el desarrollo del presente proyecto se describen en los siguientes apartados.

2.1.1. Distribuciones Linux

Para el desarrollo de este proyecto, se pueden elegir entre diferentes versiones de GNU/Linux. De este modo, y con el fin de cumplir con el requerimiento de usar el mismo sistema que el resto del equipo, se instaló Debian [2] Stretch (versión 9.2), que es la versión estable¹ de Debian. El entorno de escritorio empleado para

¹Una versión estable se refiere a aquélla que es publicada de forma oficial y que está orientada al uso de usuarios sin conocimientos avanzados de sistemas operativos y su mantenimiento.

dicho entorno es KDE, compatible con el resto de aplicaciones empleadas durante el desarrollo de este proyecto.

Por otro lado, debido a la imposición de *Nvidia* para el sistema operativo cargado en sus dispositivos, se ha usado Ubuntu[11] en su versión estable 16.04. Este sistema operativo ha sido instalado tanto en el Nvidia Jetson TX2 como en una máquina externa para realizar la compilación del código fuente y otras tareas de desarrollo.

2.1.2. Git

Con el fin de llevar el control de las diferentes versiones del desarrollo, permitiendo además la supervisión por parte del tutor del trabajo realizado, es necesario el empleo de algún tipo de “nube” virtual o repositorio en el que depositar los distintos archivos de diseño y documentación del proyecto. En nuestro caso se ha elegido el programa Git para dicho cometido.

Git es un sistema de *software* libre diseñado para el control de versiones de todo tipo de proyectos de forma eficiente [32]. De este modo, permite un desarrollo distribuido en el que los progresos llevados a cabo por distintas partes de un mismo grupo técnico se recopilan en un repositorio remoto que puede ser un servidor en la red local o a través de internet.

2.1.3. *KiCad EDA*

Existen muchos programas para diseño de placas electrónicas que son *software* propietario (concepto contrario a *open source*). Sin embargo, también existen programas para tal fin que son *software* libre, como el que se ha utilizado en la elaboración de este proyecto, *KiCad EDA* (*Electronic Design Automation*).

KiCad [31] es un programa *open-source* para la creación de esquemas electrónicos y diseño de placas de circuito impreso o *Printed Circuit Board* (PCB). Como gestor de proyectos, incorpora diferentes herramientas que trabajan con un determinado tipo de librería. Este software tiene todo lo necesario para el diseño a nivel profesional de placas de circuito impreso, tanto es así, que el *CERN* usa *KiCad* para el desarrollo de sus diseños electrónicos.

Existen dos tipos de librerías en la herramienta *KiCad EDA*: por un lado, las librerías que contienen los símbolos esquemáticos de componentes se denominan “.lib”, y las que contienen las huellas o *footprints* de dichos componentes se denominan “.mod” o “.pretty”². Las librerías por defecto de *KiCad* sólo son de

²La distinción entre estos dos tipos es que “.mod” permite agrupar el símbolo y la huella de un componente concreto como un conjunto denominado módulo, mientras que “.pretty” simplemente contiene huellas sin asociar a un determinado componente o módulo.

lectura, por lo que no pueden ser editadas; en el caso de querer añadir, algún componente, es preciso la creación de una nueva librería. Debido a lo anterior, en este proyecto se han tenido que desarrollar las librerías y huellas necesarias.

Eeschema

El diseño de esquemas eléctricos en *KiCad* se lleva a cabo a través de la herramienta *Eeschema*. Los ficheros que permiten realizar este tipo de diseños tienen la extensión “.sch”. *Eeschema* permite generar el fichero que describe las conexiones existentes entre los componentes del circuito, denominado *netlist* y con extensión “.net”. Por otro lado, la creación de nuevas librerías de tipo “.lib” se lleva a cabo en el editor de componentes de *Eeschema* llamado *LibEdit*, con el que se pueden crear nuevos componentes y librerías y/o modificar los ya existentes.

Eeschema puede tratar con tres tipos de jerarquías de esquemas. En función del tipo de jerarquía a emplear, se usan algunos comandos específicos, como el navegador de jerarquías, las hojas hijas de jerarquía y las etiquetas y/o pines correspondientes. El número de hojas de jerarquía usadas durante el diseño de un circuito eléctrico determina el tipo de jerarquía empleado.

- **Jerarquía simple:** el circuito se describe en una única hoja de esquema, por lo que sólo pueden emplearse etiquetas locales (se conectan los terminales que tengan el mismo nombre dentro de una hoja).
- **Jerarquía plana:** el esquema se describe en varias hojas de esquema al mismo nivel que deberán ser conectadas entre sí mediante etiquetas globales (conectan terminaciones que tengan el mismo nombre en distintas hojas).
- **Jerarquía compleja:** el esquema se describe en varias hojas de esquema denominadas hojas hijas que serán instanciadas mediante bloques en una hoja principal o raíz, por lo que existe un diseño a distinto nivel; se emplean etiquetas jerárquicas (conexión entre un pin jerárquico situado en un bloque instanciado y su correspondiente terminal en la hoja hija de origen).

CvPcb

CvPcb es el selector de huellas de *KiCad* que permite asociar los componentes empleados en *Eeschema* con sus *footprints*, que serán utilizados para trazar el diseño de la placa de circuito impreso. Para ello, se edita el *netlist* generado, de tal forma que se le asocie a cada componente una determinada huella de las que aparecen disponibles en las librerías.

Pcbnew

La herramienta que permite llevar a cabo placas de circuito impreso en *KiCad* es *Pcbnew*. Este *software* se emplea tras haber conformado adecuadamente el esquema del circuito en *Eeschema* y una vez asignados los *footprints* a cada componente con *CvPcb* en el *netlist*. Los ficheros que permiten realizar este tipo de diseños tienen extensión “.kicad-pcb”. La creación de nuevas librerías de tipo “.mod” se llevan a cabo en el editor de huellas de *Pcbnew* llamado *FootprintEditor*, con el que se pueden crear nuevos *footprints* y librerías y/o modificar los ya existentes. Por otro lado, *Pcbnew* incluye un visualizador del diseño mediante un modelo 3D.

Pcnew puede generar hasta 32 capas de cobre y 14 capas técnicas (serigrafía, máscara de soldadura, adhesivos, pasta de soldadura y cortes de borde), además de 4 capas auxiliares para dibujos y/o comentarios, y la visualización de conexiones virtuales entre componentes (*ratsnest*), a falta de una conexión real por medio de pistas.

Por otro lado, esta herramienta también permite obtener los ficheros necesarios para la fabricación automatizada de una placa electrónica, es decir, los GERBER del diseño. *KiCad* permite generar GERBER de tipo RS274X (impreso-fresado) y con formato 4.6 con diferentes extensiones de archivos, en función de la información que contengan.

Otro tipo de archivos que se genera en *Pcbnew* para la fabricación automática de la placa es EXCELLON, que describe los diferentes tipos de perforaciones que aloja la placa, es decir, los taladros de la tarjeta. Estos ficheros tienen extensión “.drl” y permiten describir el diámetro de los agujeros metalizados y el de los no metalizados que se llevarán a cabo de forma automatizada. Por último, también se incluye la posibilidad de imprimir fotolitos en modo transparencia y con formato “.svg” para llevar a cabo la fabricación de la placa mediante procesos químicos.

GerbView

GerbView se trata del visor de ficheros de fabricación de *KiCad*, generados en *Pcbnew*. Estos archivos pueden visualizarse conjuntamente o por separado, permitiendo una mejor apreciación de la información de la placa.

Bitmap2Component

Esta herramienta de *KiCad* permite convertir imágenes sencillas con extensión *Portable Network Graphics* o “.png” al formato de librerías de *Pcbnew*, “.mod”, con el fin de ser introducidas como en el diseño como si fueran un componente. Este útil fue pensado para introducir logotipos o imágenes representativas en las placas de circuito del usuario de *KiCad*.

2.1.4. *FreeRoute*

FreeRoute es una herramienta *online* que permite rutear automáticamente diseños de circuito impreso [5]. En ediciones anteriores de *KiCad*, existía un enlace directo a esta herramienta desde *Pcbnew*, aunque en la versión actual debe instalarse un paquete de libre adquisición en el ordenador para poder emplearse, teniendo el mismo funcionamiento.

FreeRoute trabaja mediante el intercambio de ficheros con *Pcbnew*. Para ello, primero se exporta el archivo del diseño (“.dsn”) con la configuración deseada para el trazado de pistas y éste se inicia de forma automática. Una vez terminado, se importa el archivo de sesión (“.ses”) a *Pcbnew*, concluyendo el proceso automático de trazado de pistas.

2.1.5. *Freecad*

Para el diseño de las piezas mecánicas necesarias en este proyecto, se requiere una utilidad que permita el diseño paramétrico de piezas en 3D, para este fin de ha utilizado *FreeCAD*. *FreeCAD* [4] es una aplicación libre de diseño asistido por computadora en tres dimensiones, ingeniería asistida por computadora, para la asistencia en ingeniería mecánica y el diseño de elementos mecánicos. Está basado en *Open CASCADE* y programado en los lenguajes C++ y *Python*.

FreeCAD presenta un entorno de trabajo similar a otros programas como *CATIA*, *SolidWorks*, *SolidEdge*, *ArchiCAD* o *Autodesk Revit*. Utiliza técnicas de modelado paramétrico y está provisto de una arquitectura de software modular, permitiendo añadir de forma sencilla funcionalidades sin tener que cambiar el núcleo del sistema.

A diferencia de los CAD analíticos tradicionales, como pueden ser *AutoCAD* o *Microstation*, *FreeCAD* es un CAD paramétrico, por lo que utiliza parámetros para definir sus límites o acciones. En el diseño paramétrico cada elemento del dibujo es tratado como un objeto, no estando definido únicamente por sus coordenadas espaciales (x, y, z), sino también por sus parámetros, ya sean estos gráficos o funcionales. Las bases de datos relacionadas con el objeto hacen que este software, y especialmente su banco de trabajo de arquitectura, esté muy relacionado con el enfoque *BIM*, en el que un modelo *BIM* contiene el ciclo de vida completo de la construcción, desde el concepto hasta la edificación.

2.1.6. *Inkscape*

Es necesario el uso de una herramienta que permita la creación o modificación de gráficos vectoriales. Para este fin se ha utilizado *Inkscape*. *Inkscape* [6] es un editor de gráficos vectoriales gratuito y de código libre u *open source*. *Inkscape*

puede crear y editar diagramas, líneas, gráficos, logotipos, e ilustraciones complejas. El formato principal que utiliza el programa es el *Scalable Vector Graphics (SVG)*. Este programa tiene como objetivo proporcionar a los usuarios una herramienta de elaboración de gráficos en formato vectorial escalable (SVG) que cumpla completamente con los estándares XML, SVG y CSS2. Este programa se encuentra desarrollado principalmente para los sistemas operativos GNU/Linux, pero es una herramienta multiplataforma y funciona en Windows, Mac OS X, y otros sistemas derivados de Unix.

2.1.7. *FlatCAM*

Para la creación de ficheros de trayectoria para máquinas de control numérico se ha utilizado *FlatCAM*. *FlatCAM* [3] es un software *open source* que permite llevar diseños a un enrutador CNC, o lo que es lo mismo, generar un archivo que una máquina de control sea capaz de interpretar. Este software puede abrir ficheros tipo *Gerber*, *Excellon* y generar un archivo *G-Code* que es la extensión de ficheros para este tipo de máquinas. Este software está escrito en *Python* y funciona sin problemas en la mayoría de las plataformas de modo independiente del sistema operativo.

2.1.8. *CAMotics*

Para el análisis de las trayectorias almacenadas en un fichero de trayectorias (“.gcode”), es necesario una herramienta que permita su simulación y visualización. Para tal fin se ha utilizado *CAMotics*. *CAMotics* [1] es un software *open source* que simula el fresado o grabado en una CNC de 3 ejes. Es un software de simulación rápido, flexible y fácil orientado a la comunidad *maker*³. Con *CAMotics* se puede obtener una vista previa de los resultados de las operaciones de corte antes de ejecutarlas en la máquina. Esto posibilita la comprobación del correcto trazado de las trayectorias por parte de la máquina.

2.1.9. *Slic3r*

Para la impresión de piezas 3D es necesario un *software* que permita el cálculo de las trayectorias que debe seguir la impresora. Para ello se ha utilizado *Slic3r*. *Slic3r* [10] es una herramienta necesaria para convertir un modelo 3D en instrucciones de impresión para una impresora 3D, conocidas como *G-Code*. El funcio-

³El movimiento maker surge a partir del concepto “*Do it yourself (DIY)*” ayudada por la tecnología y la difusión de internet. Está basada en el conocimiento libre, en compartir, en reciclar, en la filosofía de un mundo sostenible.

namiento básico del programa se basa en cortar el modelo en secciones horizontales, denominadas capas, para posteriormente generar trayectorias para codificarlas como *G-Code* y calcular la cantidad de material que se va a extruir.

2.1.10. *Python*

Para el desarrollo del software en el proyecto, el lenguaje de programación elegido es *Python*. *Python* [9] es un lenguaje de programación interpretado cuya filosofía hace hincapié en una sintaxis que favorezca un código legible. Se trata de un lenguaje de programación multiparadigma, ya que soporta orientación a objetos, programación imperativa y, en menor medida, programación funcional. Como hemos dicho, es un lenguaje interpretado, usa tipos dinámicos y es multiplataforma. Es administrado por la *Python Software Foundation*.

2.1.11. *OpenCV*

Para el tratamiento desde *Python* de imágenes se ha utilizado *OpenCV* [27]. Se trata de una biblioteca libre de visión artificial originalmente desarrollada por Intel. Desde que apareció su primera versión alfa en el mes de enero de 1999, se ha utilizado en infinidad de aplicaciones, desde sistemas de seguridad con detección de movimiento hasta aplicaciones de control de procesos donde se requiere reconocimiento de objetos.

2.1.12. *LATEX* y *Kile*

Para la redacción del presente proyecto se requirió un editor de textos y fórmulas, que permitiese una adecuada gestión en el sistema de control de versiones además de simplificar el trabajo de presentación. Por ese motivo, se ha utilizado *LATEX* por medio de un programa que provee de un entorno amigable para la redacción, en este caso, *Kile*.

LATEX [22, 8] es un sistema de preparación de documentos para la composición que presenta una alta calidad tipográfica. Por sus características y posibilidades, suele emplearse en la generación de artículos y libros científicos o técnicos, aunque actualmente puede emplearse casi para cualquier tipo de publicación. La finalidad de *LATEX* es que el autor del documento se centre en el contenido de éste y no es su apariencia, ya que emplea un gran número de macros que se encargan de dicha función. Así, *LATEX* está disponible como *software* libre y también es el estándar *de facto* para la comunicación de documentos científicos.

Por otro lado, *Kile* [7] es un editor de *LATEX* para escritorios de entornos KDE, también es de tipo *open source*. Entre otras utilidades, *Kile* permite autocomple-

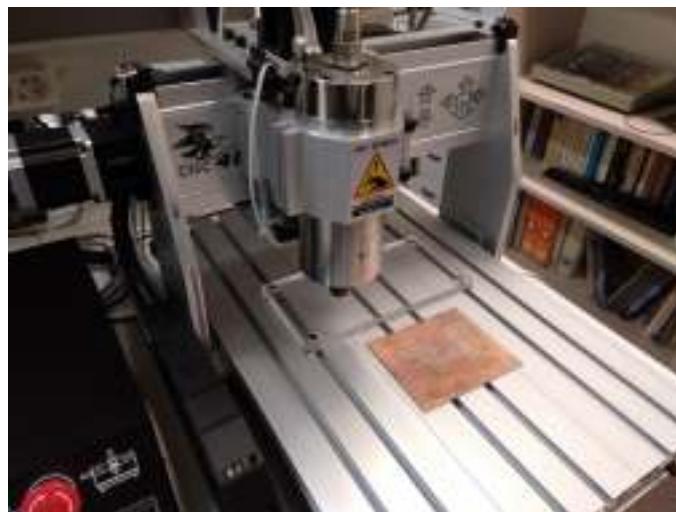


Figura 2.1: CNC-4030Z

tar comandos \LaTeX , trabajar con múltiples ficheros simultáneamente y emplear plantillas y patrones para facilitar la creación de documentos.

2.1.13. Máquinas CNC (control numérico computerizado)

Para la fabricación de las placas de circuito impreso, además de las tecnologías de fabricación clásicas con químicos, se han realizado pruebas de fabricación con máquinas de control numérico. Principalmente se han utilizado dos máquinas diferentes con distintos tipos de fresas o herramientas de corte. En la imagen de la Figura 2.1, puede verse la CNC-4030Z disponible en el laboratorio de Comunicaciones y Teledetección de la Universidad de la Laguna. Esta CNC tiene una resolución de aproximadamente $21\mu\text{m}$ en cada uno de sus ejes. Su *spindle* está refrigerado por una bomba de agua, lo cuál proporciona una buena refrigeración a la maquina durante largos periodos de trabajo. Además, la máquina cuenta con una sonda de autonivelación, lo cuál permite nivelar el eje Z de la máquina automáticamente.

Por otro lado, para las pruebas de fabricación de circuitos se dispone de la CNC-Shapeoko-3-XL, que se muestra en la Figura 2.2. Esta CNC es propiedad del *Orotava Hackerspace* y se encuentra instalada en la sede principal de esta asociación. Esta CNC está diseñada para trabajar con piezas de gran tamaño, como grandes tablones de madera. Por esta razón, la resolución de cada uno de sus ejes es de aproximadamente $30\mu\text{m}$. Al igual que la CNC de la Figura 2.1, esta máquina cuenta con autonivelador.



Figura 2.2: CNC-Shapeoko-3-XL

2.1.14. Horno de soldadura

Para la soldadura de componentes SMT se ha utilizado un horno de soldadura T962 que se muestra en la imagen de la Figura 2.3. Este dispositivo tiene una potencia de 800W lo cual le permite llegar a temperaturas de hasta 280 °C para las operaciones de soldadura.

2.1.15. Impresoras 3D

Para el prototipado rápido de algunas piezas, se ha utilizado la tecnología de impresión 3D. El modelo de impresora utilizado ha sido la Prusa i3 Hephestos que se muestra en la imagen de la Figura 2.4.

Esta impresora tiene una resolución máxima de 60 μ m en el movimiento de sus ejes. Por otro lado, se le pueden colocar diferentes tipos de boquillas para el extrusor aunque el tamaño más común es 0.4 mm. Este tipo de máquinas son capaces de imprimir piezas compuestas de materiales como PLA, HIPS, FilaFlex, PET, HDPE, y ABS entre otros.

2.1.16. Cortadora láser

Para el corte o grabado de algunas piezas elaboradas en este proyecto, se ha utilizado la tecnología de corte láser. La máquina de corte laser del Orotava Hackerspace fue utilizada para la realización de estos trabajos y puede verse en la imagen de la Figura 2.5. Este tipo de máquinas pueden cortar una gran cantidad



Figura 2.3: Horno de soldadura T962

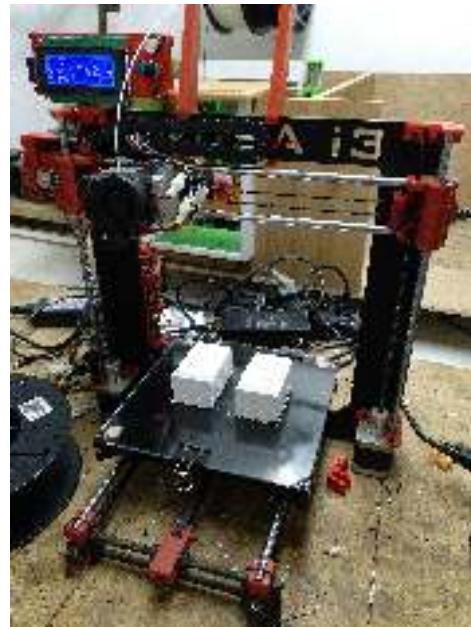


Figura 2.4: Impresora 3D Prusa i3 Hephestos



Figura 2.5: Cortadora láser del Orotava Hackerspace

de materiales, aunque principalmente se utilizan para hacer corte/grabado en madera y metacrilato.

El láser de esta cortadora tiene una potencia de 80W que pueden regularse manualmente desde su panel de control. Por otro lado, para el control de los ejes de la máquina se utiliza el puerto USB de la misma para el envío de los comandos desde un PC.

2.1.17. Lupa electrónica

Para el análisis, la soldadura y la corrección de fallos de los circuitos impresos fabricados, se ha utilizado una lupa electrónica. Gracias a la utilización de este dispositivo, se ha podido trabajar con un nivel de integración mayor de los componentes dentro del circuito. La lupa utilizada es una Lupa de Control Industrial TD-HD-708A disponibles en el Laboratorio de Comunicaciones y Teledetección y se puede ver en la imagen de la Figura 2.6.

2.1.18. Nvidia Jetson TX2

El elemento central en el desarrollo de este proyecto ha sido la Nvidia Jetson TX2. Este superordenador es un módulo basado en la arquitectura NVIDIA Pascal. Concentra toda su capacidad de cálculo en un pequeño formato de bajo consumo, idóneo para dispositivos en primera línea, como robots, drones, pequeñas cámaras y aparatos médicos portátiles. Podemos ver su kit de desarrollo en la imagen la Figura 2.7.

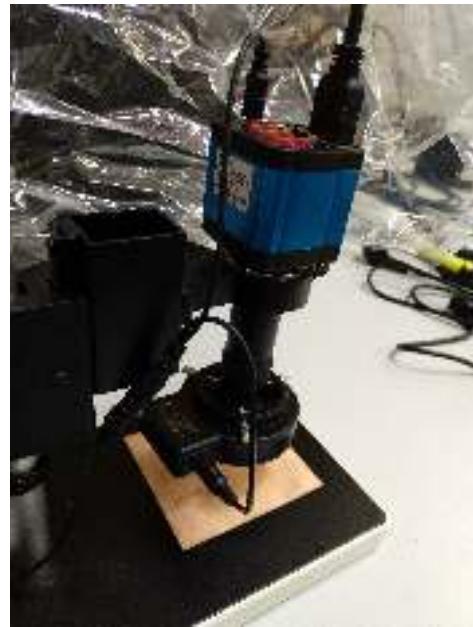


Figura 2.6: Lupa electrónica del laboratorio de Comunicaciones y Teledetección



Figura 2.7: Kit de desarrollo NVIDIA Jetson TX2

| | JETSON TX1 | JETSON TX2 |
|---------------------|---|-------------------------------------|
| GPU | Maxwell | Pascal |
| CPU | 4x 1.7GHz CPU | 64-bit Denver 2, And K27 CPU |
| Memory | 12GB DDR4 2133MHz 19.2GB | 8GB DDR4 2133MHz 9.6GB |
| Storage | 16 GB eMMC | 32 GB eMMC |
| Wi-Fi/BT | 802.11 b/g/n+BT4.2 | 802.11 b/g/n+BT Ready |
| Video Encode | H.264 up to 1080p@60 | H.264 up to 1080p |
| Video Decode | 4K H.264 | 3160p@30 |
| Camera | 1.4Gpb-21 Up to 1.5Gbps per lane | 4.4Gpb-21 Up to 2.5Gbps per lane |
| Mechanical | 4Wx1.5Hx0.5D 400g per Computer Board (no Board Connectors) | 4Wx1.5Hx0.5D |



Figura 2.8: Características de los módulos NVIDIA Jetson

Algunas características técnicas de la Nvidia Jetson se muestran en la Tabla de la Figura 2.8.

Por otro lado, este kit de desarrollo también incorpora un módulo de cámara basado en el sensor **OV5693**.

2.2. Método

Para la elaboración de este proyecto, el método de trabajo a seguir consistió en establecer los objetivos a conseguir y determinar los materiales a emplear para lograrlos, como ya se vio en la sección 1.3. Una vez hecho esto, se establecieron las etapas en las que se dividiría el proyecto, donde en cada una de ellas se desarrollarían diferentes tareas con un determinado material. De este modo, fue necesario aprender a emplear correctamente dichos materiales, intentando que fuera de la forma más autodidacta posible mediante la consulta de diferentes recursos, o bien poniendo en práctica lo aprendido durante los estudios de la titulación.

En el caso de que surgiera alguna dificultad que no pudiese ser resuelta sin ayuda, la comunicación continua con el tutor fue crucial para hallar la solución más adecuada y en el menor tiempo posible, así como para resolver dudas en cuanto a la ejecución de las diferentes tareas que componen el presente proyecto.

Por otro lado, el uso del material mencionado ha permitido una total compatibilidad a la hora de revisar el trabajo realizado, donde el uso de un repositorio Git para el proyecto ha permitido que los archivos y documentos pudieran ser com-

partidos de una forma rápida y eficaz, además de permitir un control de versiones óptimo.

Capítulo 3

Diseño e integración del sistema

Para el diseño y la fabricación del sistema de adaptación multicámara, es necesaria la integración de una serie de elementos que permitan la adquisición y el tratamiento de imágenes. En este capítulo, se introducen de forma conceptual los diferentes módulos que componen el sistema además de como se integran para configurar el sistema completo.

3.1. Descripción general del sistema

En el diagrama de bloques de la Figura 3.1, pueden observarse los elementos que componen el sistema. En primer lugar se encuentra una estructura mecánica de soporte y dentro de esta encontramos el sistema de cámaras denominado *array*. Este *array* está compuesto por **6 cámaras CSI-2¹**. El *array* dispone de un **sistema de calibración mecánica** que permite regular el ángulo de apuntado de las cámaras. Estas se conectan a una placa de desarrollo de Nvidia denominada **Nvidia Jetson TX2(NJTX2)**. La interfaz de conexión entre el *array* de cámaras y la NJTX2 se hace por medio de lo que denominamos **módulo puente** que permite transmitir los datos entre las cámaras y la placa. El sistema operativo de la NJTX2 lee y envía datos a las cámaras a través del **módulo puente** y debe ser configurado mediante un arbol de información de periféricos que se denomina **device tree** y que permite que el *kernel* pueda realizar esta tarea. Por último, se encuentra el sistema de adquisición y tratamiento de imágenes con el que se procesarán las imágenes capturadas para extraer la información 3D de las escenas.

Por otro lado, en la misma situación que el módulo puente en la placa de desarrollo, en el conector J22, se ha diseñado una variante denominada **módulo**

¹El estándar CSI o Interfaz Serie para Cámaras (del inglés, Camera Serial Interface) es una especificación de la Mobile Industry Processor Interface Alliance (MIPI Alliance) que define la interfaz entre una cámara digital y un procesador anfitrión.

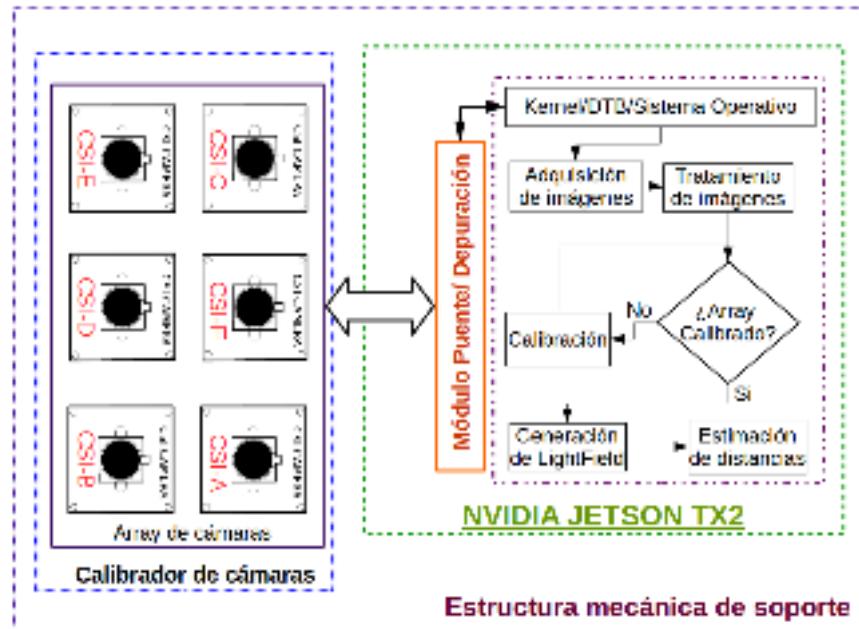


Figura 3.1: Diagrama de bloques de elementos del sistema

de depuración que permite el análisis de las líneas procedentes de la NJTX2. Este módulo no se integra dentro del funcionamiento operativo del sistema, sino que se ha diseñado para ser utilizado en la comprobación de las distintas líneas salientes del conector J22 de la NJTX2 y por ello para ser usado durante el proceso de desarrollo del sistema.

3.2. Descripción de elementos del sistema

En las siguientes secciones se describe cada uno de los elementos que componen el sistema.

3.2.1. Array de cámaras

El *array* de cámaras está compuesto por varias cámaras que transmiten sus datos usando el protocolo CSI-2². Para su soporte, se utilizó la denominada estructura de calibración, que además de permitir el ajuste del apuntado de las cá-

²MIPI CSI-2 es la interfaz de cámara más utilizada en la industria móvil. Ha logrado una adopción generalizada por su facilidad de uso y su capacidad para admitir una amplia gama de aplicaciones de alto rendimiento, incluyendo 1080p, 4K, 8K y más, además de fotografía de alta resolución.

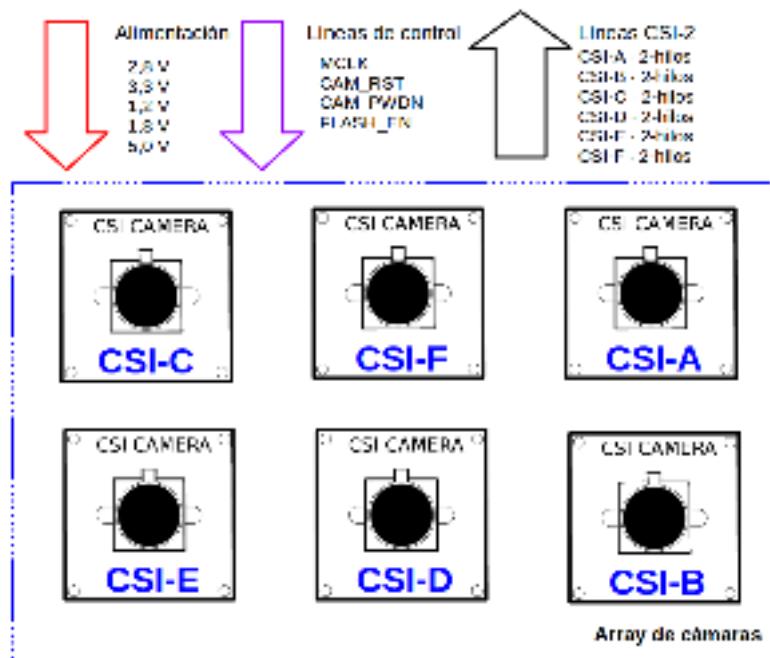


Figura 3.2: Diagrama de bloques del flujo de datos del *array* de cámaras

maras, permite mantener la disposición espacial de las cámaras. En el diagrama de bloques de la Figura 3.2 puede verse la disposición del *array* de cámaras y el flujo de datos.

En este diagrama puede observarse que todas las cámaras transmiten sus datos en líneas CSI-2 separadas y que por contra, para su manejo se usan las mismas señales de comando sin distinción de la cámara. Para lograr un control individualizado de cada dispositivo, se realiza una multiplexación de las señales de control, que en este caso se envían mediante i^2c . Esta multiplexación se debe realizar en el **módulo puente**.

Con respecto a las alimentaciones, las cámaras disponen de varios niveles de voltaje en función del sensor que se utilice, normalmente esta tensión es 3,3V para la mayoría de sensores comerciales. Además, es común utilizar una tensión de 2,8V para dispositivos digitales y otra tensión de 1,8V para la alimentación de dispositivos analógicos como el obturador del sensor de cámara.

Calibrador de cámaras

Es importante que las cámaras que estén colocadas en el *array* tengan un punto en el cuál coincidan las direcciones de apuntado, simplificando la aplicación de algoritmos para el cálculo de mapas de distancia.

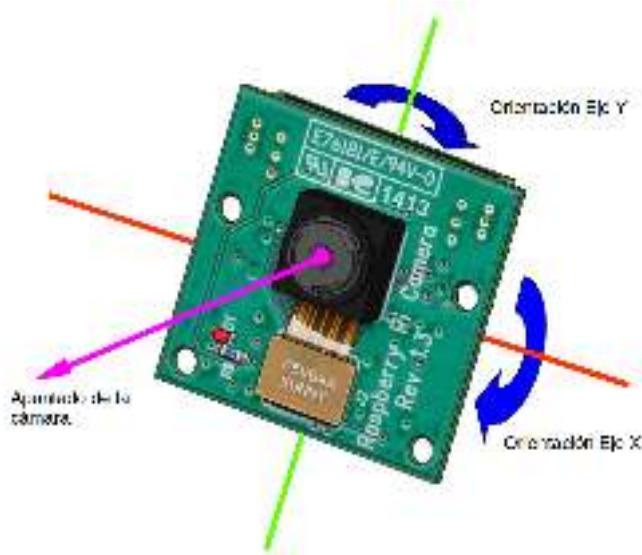


Figura 3.3: Giros para el cambio de orientación de los sensores de cámara

Las cámaras disponen de un soporte que permite su sustentación, que también permite regular su ángulo de apuntado. Para la fijación de las cámaras al soporte se utiliza un tornillo y dos más para el ajuste del apuntado. En la Figura 3.3, se pueden observar los ángulos de giro que tiene un módulo de cámara al usar el calibrador. Se observa que las rotaciones que se pretenden controlar coinciden con las direcciones vertical y horizontal de la imagen. Además, existe un grado de libertad más, se trata de la rotación sobre el eje que es paralelo a la dirección de apuntado de la cámara. Asumiendo que las cámaras están alineadas con respecto a sus ejes una vez fabricadas, el ángulo de rotación con respecto a este último eje (el eje paralelo a la dirección de apuntado) será el mismo para todas y no será necesario considerar este grado de libertad en el diseño.

3.2.2. Nvidia Jetson TX2

La estructura encargada de la adquisición de los datos y el procesamiento de las imágenes es la NVIDIA Jetson TX2. Este dispositivo dispone de un amplio número de características y periféricos que permiten la interacción con módulos externos, en el caso del presente proyecto se utilizan los puertos que permiten la lectura de líneas CSI-2, es decir, su conector J22. Además de la posibilidad de realizar la lectura directa desde las líneas CSI, la NJTX2 dispone de una GPU³

³La unidad de procesamiento gráfico o GPU (Graphics Processing Unit) es un coprocesador dedicado al procesamiento de gráficos u operaciones de coma flotante, para aligerar la carga de

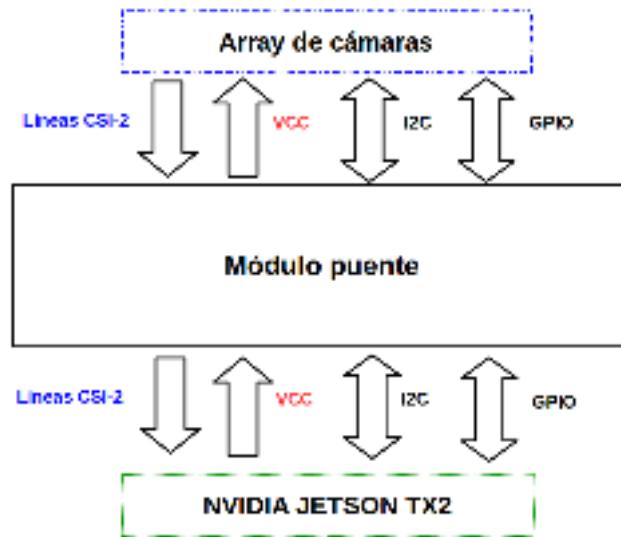


Figura 3.4: Diagrama de señales del módulo puente

que permite un procesamiento de gráficos más rápido entre otras cosas.

En las siguientes secciones se explicarán en mayor detalle los elementos que permiten la adquisición y lectura de las imágenes. Por otro lado, también se introducen los elementos de *software* que permiten el tratamiento de las imágenes.

Módulo puente/Depuración

Para la adquisición de los datos por medio de las líneas CSI que salen del *array* de cámaras (véase el diagrama de la Figura 3.2) se utiliza el **módulo puente** del que puede verse un diagrama en la Figura 3.4.

El módulo puente hace de interfaz de paso para las conexiones entre el *array* de cámaras y la NJTX2 y debe gestionar los datos que se envían en ambas direcciones. La principal dificultad que debe afrontar el módulo puente es la poca cantidad de líneas i^2c para el control de las cámaras disponibles en la NJTX2. Para realizar esta tarea la NJTX2 dispone de cuatro buses para el control de cámaras, aunque en la práctica se dispone únicamente de tres de ellos. La solución adoptada para el correcto control de múltiples cámaras conectadas a los buses i^2c se basa en el uso de un traductor de direcciones. Puede verse un diagrama en la Figura 3.5. Esta traducción de direcciones consiste en cambiar las direcciones i^2c de las cámaras de tal forma que puedan estar dos cámaras con la misma dirección i^2c en un mismo bus. Para más información véase el anexo C.5 [14].

trabajo del procesador central en aplicaciones como los videojuegos o aplicaciones 3D interactivas.

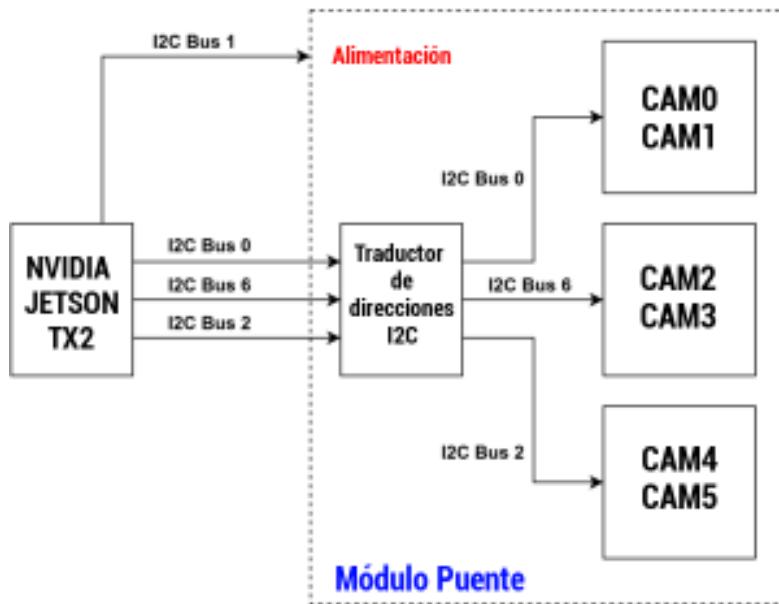


Figura 3.5: Diagrama de traducción de direcciones en módulo puente

Por otro lado, para el análisis de las líneas procedentes del conector J22 de la NJTX2, se utilizará lo que se ha denominado **módulo de depuración**. Este módulo ha sido diseñado para disponer de una interfaz de conexión simple entre la NJTX2 y un equipo de medida o análisis, como podría ser un multímetro o un analizador lógico.

Kernel y device tree

Para detectar el hardware añadido a la NJTX2 a través de conector J22 (véase el Anexo D) es necesaria la configuración del **kernel** y de lo que se conoce como DT o *Device tree*, ya que la NJTX2 dispone de un sistema operativo Linux. Un *Device tree* es una estructura de datos que describe los componentes de hardware de un dispositivo particular para que el *kernel* del sistema operativo pueda usar y administrar esos componentes, incluyendo la CPU o GPU, la memoria, los buses y los periféricos, y en el caso del presente proyecto las cámaras CSI y el módulo puente. Los periféricos como las cámaras CSI no tienen la posibilidad de ser detectados de forma automática por la NJTX2 sin especificar explícitamente el puerto en el que serán conectadas, sus velocidades de reloj, número de líneas, etc. Por esta razón, es necesaria la configuración del *Device tree*.

En el diagrama de la Figura 3.6 se muestra la relación entre la conexión física de las cámaras a la NJTX2. Para la correcta detección de los elementos de *hard-*

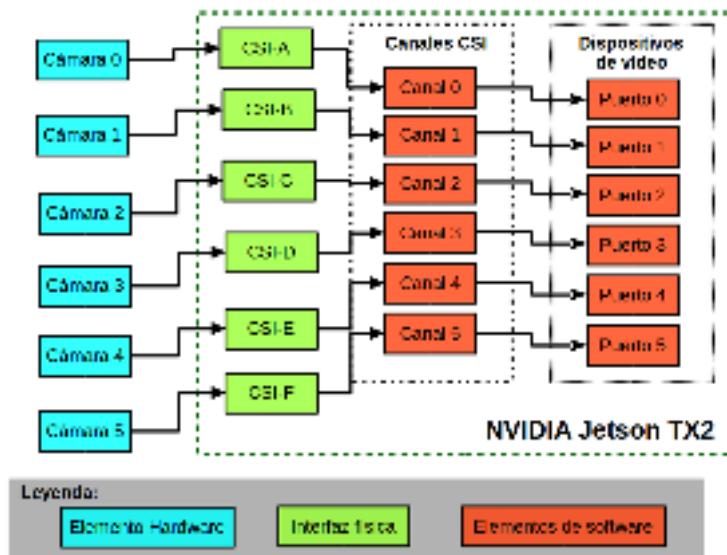


Figura 3.6: Diagrama de relación entre las cámaras y los elementos software del *device tree*

ware del diagrama por parte de la plataforma, es necesaria la configuración de los elementos de *software* mostrados por medio del *device tree*.

Por otro lado, una vez realizada la configuración del *device tree* es necesario configurar y recompilar el **kernel** de la NJTX2 para que puedan usarse los dispositivos que se han conectado a la misma. Además de la configuración del *kernel* es necesario desarrollar módulos y drivers que se compilen como parte del sistema operativo y que permitan el uso de los periféricos que se conecten al sistema. En el caso de los sensores de cámara, los módulos y drivers permitirán el uso de la cámara por parte del sistema operativo al igual que ocurre con una *webcam*.

Software de procesado de imágenes

Para el procesado de las imágenes adquiridas por medio de los recursos del sistema operativo, como por ejemplo *v4l2*⁴, son necesarias una serie de utilidades. Estas utilidades cumplen unos objetivos muy concretos como por ejemplo el ajuste del color o la eliminación de ruido de las imágenes. Además, se debe desarrollar utilidades que permitan la calibración del *array* de cámaras y adicionalmente una herramienta que permita el cálculo del mapa de distancias partiendo de las 6 imágenes obtenidas por las cámaras.

⁴*Video4Linux*, V4L2, para abreviar, es una colección de controladores de dispositivos y una API para realizar la captura de video en tiempo real en sistemas Linux.

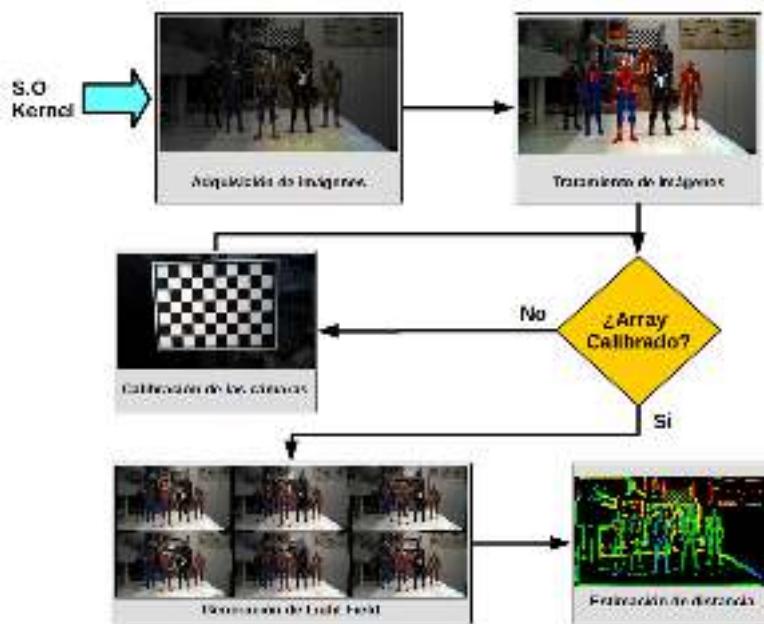


Figura 3.7: Adquisición y tratamiento de imágenes

En el diagrama de la Figura 3.7 puede observarse el flujo seguido por los datos de las imágenes y las etapas de procesado. La primera etapa de procesado de las imágenes es especialmente relevante debido a que los datos *raw* devueltos por el sensor no representan de manera fiel la escena que se ha capturado si no se han tratado previamente. Una vez tratados los datos se pasará a una etapa de calibración, en el caso de que el *array* de cámaras no esté calibrado. La calibración implica conocer los parámetros intrínsecos y extrínsecos de las cámaras además de asegurar que sus direcciones de apuntado coincidan en un punto, como ya se comentó en secciones anteriores. Una vez realizado el proceso de calibración, se generará el *light field* y como paso final del proceso se generará el mapa de estimación de disparidad.

3.2.3. Estructura mecánica de soporte

Para el soporte de todo el sistema, en el diagrama de la Figura 3.1 se indica la existencia de una estructura mecánica de soporte. Esta estructura permite el montaje de toda la electrónica necesaria para el correcto funcionamiento de la NJTX2 y el sistema de cámaras. Por otro lado, también permite el montaje del *array* de cámaras con su estructura de calibración incorporada. Adicionalmente, la estruc-

tura permite ajustar la posición espacial del *array* de cámaras y la electrónica que lo acompaña.

Capítulo 4

Diseño de la electrónica

En este capítulo se muestra el diseño electrónico que se ha elaborado para la adaptación de los módulos de cámara en la Nvidia Jetson TX2. Por otro lado, se introducen los diseños realizados para el módulo puente y el módulo de depuración. En las siguientes secciones se introducen cada uno de los módulos diseñados.

4.1. Módulos principales a diseñar

El desarrollo electrónico se ha realizado principalmente para la adquisición de imágenes por medio de un sensor CMOS a través de las líneas CSI-2 del mismo. Además, se diseña la interfaz que conecta los módulos de cámara con la NJTX2. Por esta razón, los diferentes módulos que se desarrollan tienen una serie de bloques relacionados entre sí.

Para conseguir el circuito deseado, se han tenido en cuenta las referencias dadas por Nvidia en sus manuales para el desarrollo de productos [14]. Principalmente se han desarrollado los diseños de los módulos de cámara y lo que se ha denominado módulo puente. Adicionalmente, se ha desarrollado un módulo depurador de líneas para tener la posibilidad de analizar las señales de la Nvidia Jetson TX2 con ayuda de un analizador lógico.

En la imagen de la Figura 4.1, se puede observar el diagrama de bloques del módulo puente que conectará el módulo de cámara con la NJTX2. En el caso de incorporar varios sensores de cámara (hasta un máximo de 6), el módulo puente deberá tener una EEPROM y un multiplexor i^2c para poder administrar todas las direcciones i^2c de todos los módulos de cámara.

Por otro lado, en la Figura 4.2 se puede ver el diagrama de bloques del diseño del módulo de cámara.

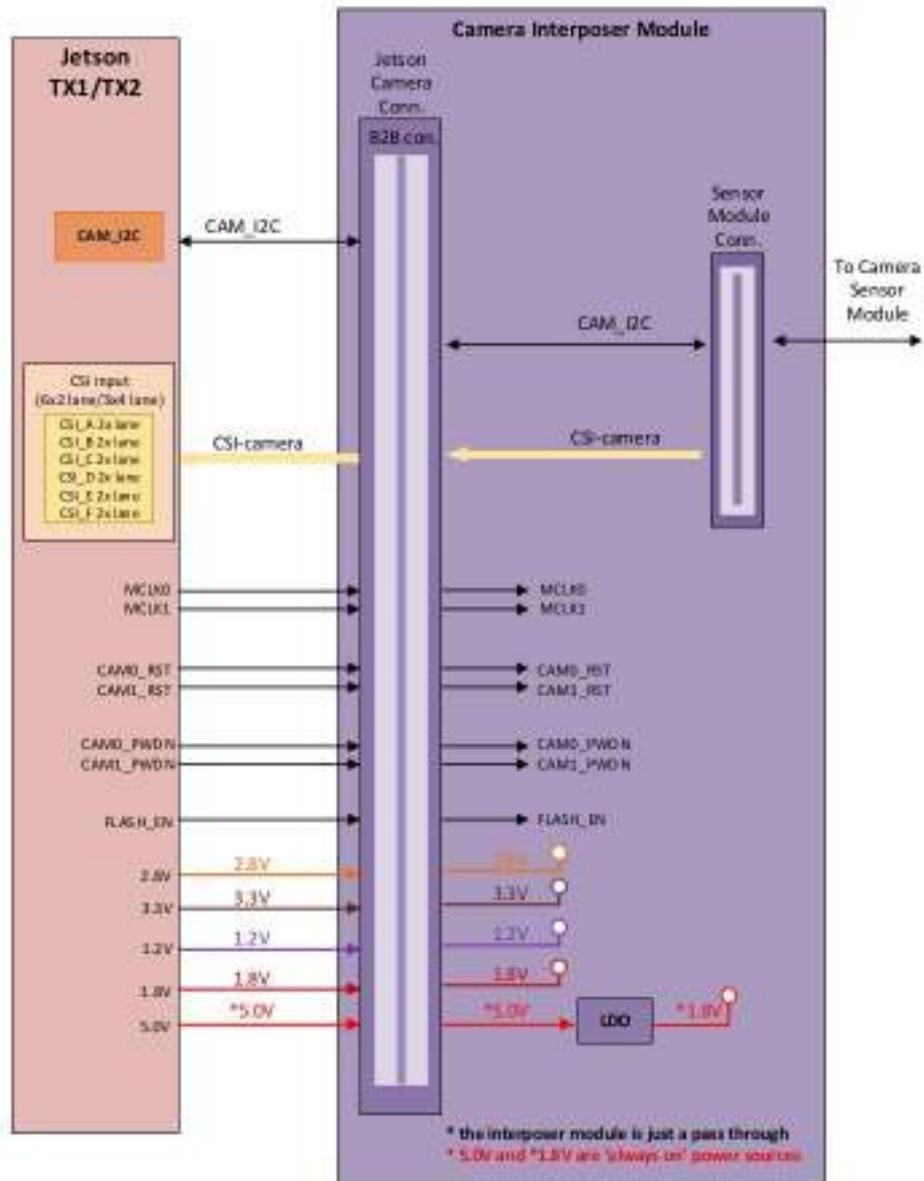


Figura 4.1: Diagrama de diseño del módulo puente

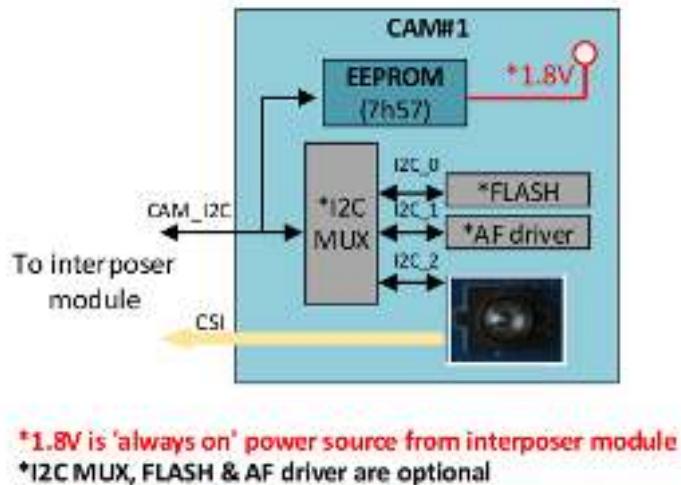


Figura 4.2: Diagrama de diseño del módulo de cámara

En las siguientes secciones se analizan cada uno de los módulos y sus componentes fundamentales.

4.1.1. Módulo de cámara

El elemento principal del diseño es el denominado como módulo de cámara. Este módulo debe ser capaz de adquirir imágenes desde un sensor CMOS además de disponer de una EEPROM para su identificación y direccionamiento dentro del bus i^2c . El módulo de cámara también debe disponer de la electrónica necesaria para evitar posibles ruidos en las líneas de transmisión del CSI.

Con respecto al sensor CMOS utilizado en este módulo, inicialmente se han diseñado los módulos de cámara para el sensor OV5680 aunque posteriormente se ha estudiado otros sensores. Pueden consultarse las hojas de características de los sensores que se han estudiado en este trabajo en el Anexo E.

4.1.2. Módulo puente

El módulo puente como ya se ha dicho, será el responsable de servir de interfaz de conexión entre la NJTX2 y el módulo de cámara que se utilice. Desde la NJTX2 se conectarán al módulo puente las diferentes alimentaciones que sean necesarias para el módulo de cámara y que pasarán a su través. Las alimentaciones de salida del conector J22 de la Nvidia Jetson se encuentran en la Tabla 4.1.

| Nombre de la conexión | Especificaciones | Notas |
|-----------------------|----------------------|--|
| VDD 5V0 | 5.0V | Esta señal se encuentra siempre encendida ¹ |
| VDD 3V3 SLP | 3.3V, 4A (máximo) | 3.3V Voltaje del sistema |
| AVDD CAM | 2.8V, 3A (máximo) | 2.8V Voltaje analógico |
| DVDD CAM 1V8 | 1.8V, 2A (máximo) | 1.8V Voltaje digital de entrada/salida |
| DVDD 1V2 | 1.2V, 300mA (máximo) | 1.2V Voltaje digital |

Tabla 4.1: Alimentaciones disponibles en el conector J22.

Además, se conectarán las líneas CSI que fueran necesarias, para los módulos de cámara usados en este proyecto se utilizarán líneas CSI de dos hilos. También se conectan las señales de reloj, y los GPIO² que servirán de señal de *reset* y *powerdown*. Adicionalmente se podrían utilizar otros GPIO para la conexión del auto-enfoque, la activación del flash, el control del obturador o alguna otra funcionalidad extra del módulo de cámara.

Por todo ello, el módulo puente juega un papel fundamental en el desarrollo del proyecto ya que concentrará todas las señales y las conexiones entre elementos.

4.1.3. Módulo de depuración

Este módulo extra, no aportará ninguna funcionalidad al funcionamiento del *array* de cámaras. Sin embargo, servirá para estudiar el comportamiento de las diferentes líneas del conector J22 de la Nvidia Jetson. Este módulo, se diseñará buscando la posibilidad de medir con un analizador lógico el comportamiento de las distintas líneas procedentes de la NJTX2.

4.2. Restricciones de diseño

Hay que tener en cuenta que para integrar una placa de circuito impreso en un dispositivo ya fabricado como la Nvidia Jetson TX2, hay que cumplir con una serie de restricciones físicas en las dimensiones de los módulos. De no ser así, lo más probable es que el circuito que se diseñe no encaje en los agujeros de montaje de la Nvidia Jetson y no se pueda integrar con éxito el diseño.

En el caso del módulo de cámara no se contempla ninguna restricción en cuanto al tamaño de la PCB que se diseña, por contra, Nvidia impone unas dimensiones para los módulos puente que quieran ser incorporados a la NJTX2. Esto es así ya

¹Las alimentaciones se gestionan desde el sistema operativo de la NVIDIA Jetson

²General Purpose Input/Output, Entrada/Salida de Propósito General

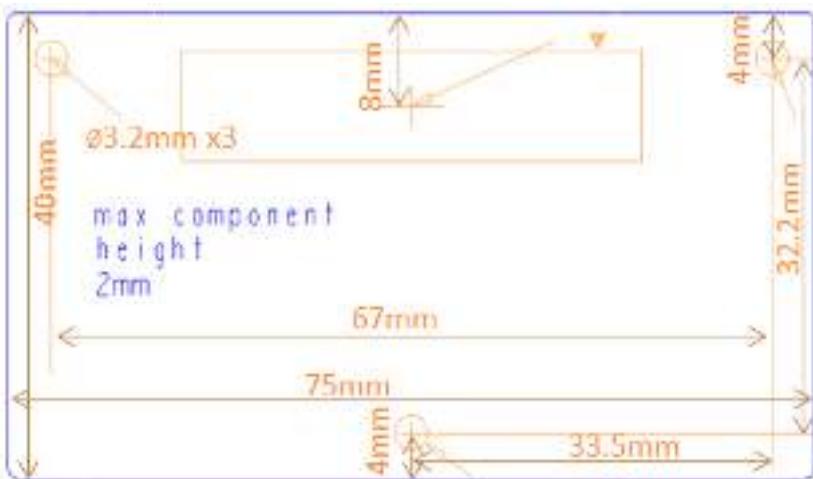


Figura 4.3: Restricciones geométricas del módulo puente

que al ser una placa de tan reducido tamaño dispone de poco espacio para la incorporación de dispositivos externos. Las restricciones para las dimensiones del módulo puente pueden verse en boceto de la Figura 4.3. Por ello, estas restricciones geométricas deben ser respetadas.

Por otro lado, en el caso del módulo de depuración, se han llevado al límite las restricciones geométricas de la placa procurando que los agujeros de montaje disponibles en la Nvidia Jetson TX2 sirvieran para la fijación del módulo.

4.3. Direcciónamiento i^2c

Además de las restricciones mecánicas, Nvidia también indica cuál deberá ser la dirección i^2c de los dispositivos que se desarrollen, en caso del módulo puente y el módulo de cámara [14]. Además también indican cuál debe ser la dirección i^2c de cada uno de los integrados dentro del bus. En la Tabla 4.2 se indican cuales son las direcciones i^2c de cada uno de los dispositivos. Es importante tener en cuenta las direcciones i^2c de cada uno de los componentes tanto en la fase de diseño como en la fase de programación de los drivers y el diseño de la interfaz *hardware-software* como se verá más adelante. Esto es así debido a que la presencia de las EEPROM con esas direcciones i^2c permiten a la NJTX2 detectar dinámicamente en su encendido, el módulo puente y los módulos de cámara. Como consecuencia, podrá cargar los correspondientes módulos y *drivers*.

| Jetson TX2 | Módulo puente | Dirección | Sensores de cámaras | Dirección |
|------------|------------------------------|-----------|------------------------------|-----------|
| CAM I2C | EEPROM, AT24C02D | 0x54 | EEPROM, AT24C02D | 0x57 |
| | I2CMUX, TCA9546A | 0x70 | Flash driver IC (opcional) | ND |
| | I2C GPIO Expander (opcional) | ND | Auto focus driver (opcional) | ND |

Tabla 4.2: Direccionamiento i^2c de integrados para la Nvidia Jetson TX2

4.4. Metodología de diseño

En los siguientes apartados se introducirá el método de desarrollo del diseño de la electrónica y el trazado de las PCB.

4.4.1. Creación de componentes

Para la creación de nuevos símbolos esquemáticos se necesita indicar el nombre del componente y su referencia (tipo de componente). Fundamentalmente, cada esquemático consiste en un espacio delimitado con cierta cantidad de pines numerados y de función específica, debiendo ser éstos últimos totalmente acordes con lo indicado en el *datasheet* del componente. Los esquemáticos fueron obtenidos a partir de editar los existentes en el resto de librerías, para el caso de los más complejos, o se crearon desde cero, para el resto de casos.

4.4.2. Creación de huellas

La creación de nuevas huellas de componentes conlleva indicar el nombre por el que serán buscados en la librería (suele ser el tipo de encapsulado), así como su referencia, que será asociada al símbolo esquemático del componente. Fundamentalmente, cada huella consiste en un espacio delimitado con cierta cantidad de pads de tipo SMT (*Surface Mount Technology*) o THT (*Through-Hole Technology*), que son definidos por su tamaño, forma y distancia de separación entre sí. Esta información viene dada en las especificaciones del *datasheet* del componente al que se le relaciona o, en su caso, en el estándar apropiado. Los pads deben estar numerados correctamente para que la asociación con los pines del símbolo esquemático sea correcta.

Las huellas fueron obtenidas a partir de editar las ya existentes en el resto de librerías, para el caso de las más complejas, o se crearon desde cero, para el resto de casos.

4.4.3. Diseño esquemático

La realización de esquemas consiste en ubicar componentes (tanto los de las librerías por defecto, como los editados o nuevos guardados en las librerías de usuario) dentro de una o varias hojas de esquema y unirlos mediante hilos o etiquetas.

Una vez conformado el esquema, se debe llevar a cabo la anotación o numeración de todos los componentes del diseño. Esta operación consiste en asignar un valor numérico a la referencia de letras que indican el tipo de componente, con el fin de distinguir unos elementos de otros y agrupar los que pertenezcan a una misma parte. La anotación puede llevarse a cabo manualmente (secuencial) o automáticamente (simultánea) mediante la herramienta *Annotate*. En el caso del presente proyecto, se llevó a cabo una numeración manual.

Tras realizar este paso, se debe comprobar que todos los componentes están anotados y sus pines conectados correctamente, es decir, se comprueba que el diseño cumpla con ciertas normas eléctricas mediante la herramienta *Electrical Rules Check* (ERC). Algunas de estas verificaciones vienen activadas por defecto, pero pueden ser modificadas o desactivadas en la matriz ERC, que es la que detecta las incongruencias a nivel eléctrico de los esquemas. En el caso del presente proyecto, no fue necesario modificar dicha matriz, ya que los errores que se mostraban en el reporte fueron sencillos de resolver.

Por último, el paso previo al diseño de la PCB consiste en generar el *netlist*. Este proceso puede realizarse de dos maneras a la hora de emplear un determinado componente en el esquema, indicando en ese momento su *footprint* correspondiente, o tras completar el esquema mediante la herramienta CvPcb, que permite visualizar todas las huellas disponibles. En el presente proyecto, se han llevado a cabo ambos procedimientos.

4.4.4. Diseño de circuito impreso

El diseño de circuito impreso de la placa comienza por la lectura del *netlist* desde *Pcbnew*, que permite exportar la información que este archivo contiene en cuanto a huellas y conexiones. Una vez hecho esto, aparecerán agrupadas todas las huellas y sus conexiones entre sí (*ratsnest*) en el entorno de trabajo, permitiendo empezar con el emplazamiento de componentes de acuerdo al diseño real.

Normalmente, la distribución de componentes se lleva a cabo de tal manera que el ruteo o trazado de pistas sea el más corto posible, disminuyendo el número de pérdidas en la transmisión de señales. Este hecho se ha tenido en cuenta a la hora de emplazar todos los componentes dentro de las placas.

Las pistas de un diseño suelen trazarse manualmente con las herramientas de *Pcbnew* y procurando que el ancho de pistas sea siempre el mayor posible, tal

y como se ha llevado a cabo. A la hora de trazar las pistas, se realiza en todo momento el *Design Rules Check* (DRC), que comprueba continuamente que las dimensiones definidas en la configuración de ruteo de la placa se cumplan. Cuando por fin se consigue el trazado de las pistas en la placa sin errores, se pueden establecer los planos de cobre, tanto de masa como de alimentación, en las láminas conductoras de la placa (cuantas más capas de cobre haya, mayor será el número de planos definidos).

Una vez obtenido el diseño completo de circuito impreso, es preciso delimitar los bordes de la placa y prever el uso de agujeros mecánicos de sujeción que permitan acoplar ésta a algún tipo de estructura. Por otro lado, también puede añadirse algún tipo de texto o logotipo identificativo de la placa, tal y como se ha llevado a cabo en este proyecto.

Por último, y para finalizar con el diseño de la placa, es necesario generar los archivos apropiados que contengan toda la información necesaria para fabricar adecuadamente la tarjeta. Estos archivos dependerán del tipo de fabricación que usemos para las placas.

4.5. Diseño preliminar

Para evaluar la posible viabilidad de los módulos diseñados se han planteado varios diseños preliminares, que posteriormente se valorarán y se decidirá sobre la viabilidad de fabricación de dichos módulos.

En el diseño preliminar de los módulos, estos conservan gran parte de los componentes recomendados por Nvidia en su manual para el desarrollo de módulos de cámara [14]. Por otro lado, aquellos componentes que no resulta posible integrar por su alta exigencia de capacidad de fabricación, se han sustituido por equivalentes o por circuitos que realizan la misma función.

4.5.1. Consideraciones de diseño previas

Para realizar los diseños de los diferentes módulos se han tenido en cuenta una serie de consideraciones comunes en cuanto a las tensiones de alimentación de los diferentes circuitos, el direccionamiento i^2c en las configuraciones de las EEPROM y el filtrado de las interferencias electromagnéticas.

Estabilización de tensiones continuas de alimentación

Para asegurar que la tensión de alimentación de los módulos se mantiene constante y además está libre de ruidos, se propone el uso de los circuitos que se muestran en la Figura 4.4

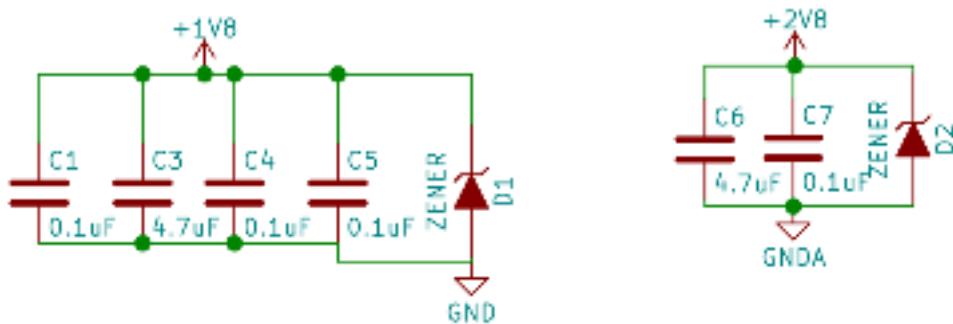


Figura 4.4: Circuitos para la adecuación de las alimentaciones

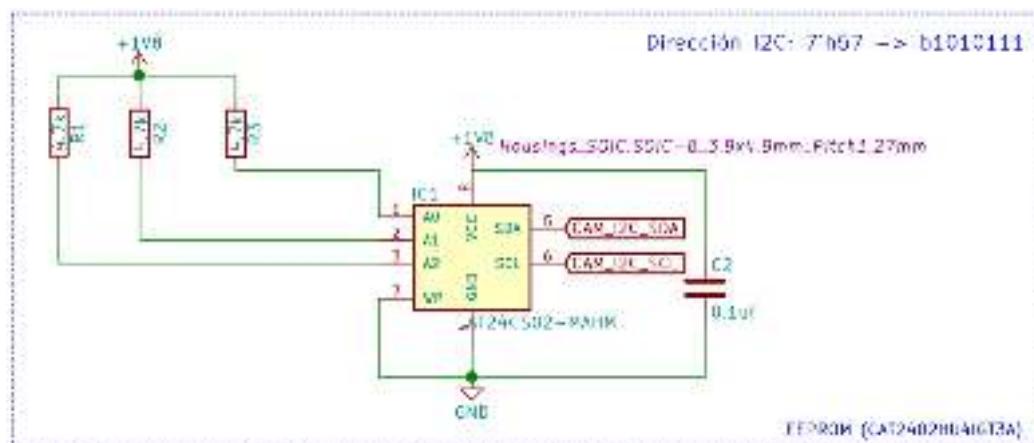


Figura 4.5: Configuración de EEPROM para el módulos de cámara

Gracias a la presencia del diodo Zener, se asegura que la tensión de alimentación se mantiene constante. Por otro lado, por la presencia de los condensadores, se absorbe parte del ruido de la señal y además se aporta corriente en el caso de que la carga conectada a la alimentación así lo demande.

Configuración de EEPROM para el direccionamiento i^2c

Como se ha visto en la sección 4.3, uno de los requerimientos en el diseño de los diferentes módulos serán las direcciones i^2c de los integrados que se incorporen al diseño. Por ello, se propone la configuración de la Figura 4.5 en el caso de que se coloque en el módulo de cámara. Por contra, si la EEPROM se coloca en el módulo puente su configuración será la observada en la Figura 4.6.

Ambas configuraciones se han diseñado para cumplir con las especificaciones de la Tabla 4.2. Para más información sobre las EEPROM utilizadas y las posibles configuraciones de direccionamiento véase el Anexo B.3.

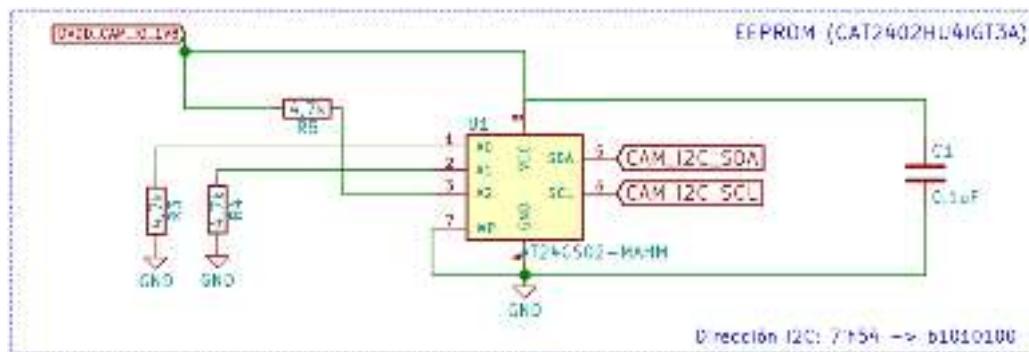


Figura 4.6: Configuración de EEPROM para el módulos puente

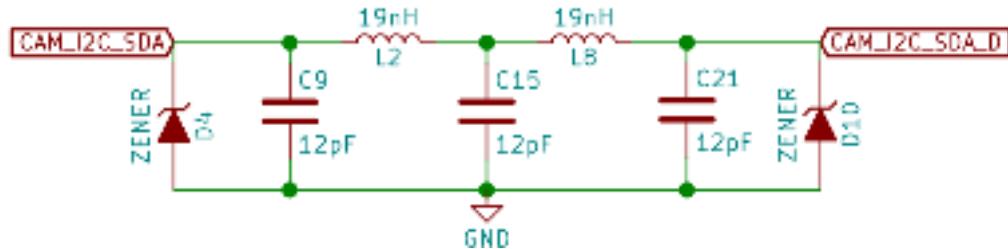


Figura 4.7: Filtro para interferencias electromagnéticas

Filtrado de interferencias electromagnéticas

Para el filtrado del ruido en las líneas de datos, se utilizarán filtros EMI³. El integrado elegido para realizar el filtrado es el *EClamp2398p* cuyo *datasheet* se muestra en el Anexo B.4. Sin embargo, debido al reducido tamaño del encapsulado de este componente su integración en una placa de circuito impreso en el momento de la fabricación tiene una dificultad demasiado elevada. Por esta razón, se ha buscado una alternativa. Dicha alternativa es el circuito del diagrama mostrado en la Figura 4.7, cuyo funcionamiento es el mismo que el del *EClamp2398p*.

4.5.2. Módulos de cámara diseñados

Con respecto a los módulos de cámara, en este proyecto únicamente se ha diseñado un módulo de cámara basado en el sensor OV5680. El resto de módulos sobre los que se ha trabajado han sido módulos con su propia electrónica incorporada, por lo que no ha sido necesario diseñarla. Los otros dos módulos de cámara que se han utilizado en los diseños son los basados en los sensores OV5647 e IMX219. Ambos sensores se han utilizado para desarrollar módulos de cámara para Raspberry Pi que pueden ser integrados en la NJTX2. Pueden verse

³Electromagnetic interference filter

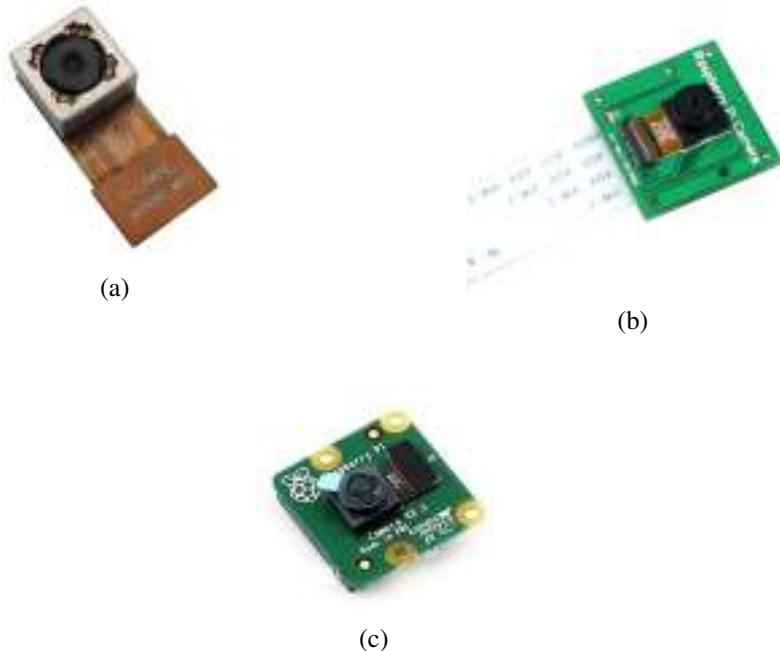


Figura 4.8: Sensor y módulos de cámara utilizados

los sensores de cámara utilizados para el diseño en la Figura 4.8. Además, puede consultar todos los módulos estudiados durante la etapa de diseño en el Anexo E.

Módulo de cámara basado en el sensor OV5680

Los componentes a utilizar en el módulo de cámara basado en el sensor OV5680 son los que se observan en la Tabla 4.3:

| Grupo | Componente | Valor | Encapsulado | Unidades |
|------------------|---------------------|------------------------------|------------------------------------|----------|
| Condensadores | C1206C104J5RACTU | 100nF | 1206 (in) | 5 |
| | C3216X7R1E475K160AC | 4.7 μ F | 1206 (in) | 2 |
| | C1206C120J5GACTU | 12pF | 1206 (in) | 18 |
| Módulo de cámara | OV5680 | - | - | 1 |
| Resistencias | CRG1206F4K7 | 4.7k Ω F | 1206 (in) | 3 |
| Conector | Molex | Serie FFC/FPC 30 PIN | SMT | 1 |
| Bobinas | HK212518NK-T | 19nH | 0603 (in) | 12 |
| Diodos ZE-NER | 1SMA5917BT3G | 1.5 W 2.5 μ A 5 Ω | SMA (in) | 14 |
| EEPROM | AT24C02D-SSHM-T | EEPROM 1.7V | TSSOP-8 4.4x3mm Pitch 0.65mm | 1 |

Tabla 4.3 – continua en la página siguiente

Tabla 4.3 – *continuación de la página anterior*

| Grupo | Componente | Valor | Encapsulado | Unidades |
|-------|------------|-------|-------------|----------|
|-------|------------|-------|-------------|----------|

Tabla 4.3: Componentes del módulo de cámara OV5680

El funcionamiento de este módulo es simple. Las líneas CSI procedentes del sensor, pasan por una serie de filtros y se envían al conector que irá al módulo puente. Por otro lado, las alimentaciones procedentes del módulo puente, se pasarán por una serie de filtros (véase la sección 4.5.1) para estabilizar la señal de continua de alimentación y así evitar posibles picos de ruido. Para ello se han empleado las resistencias, condensadores, bobinas y diodos zener del circuito. Por otro lado, hemos utilizado un conector Molex de 30 pines para la conexión entre el módulo de cámara y el módulo puente (Anexo B.1). Por último, otro elemento que juega un papel primordial en este diseño es la EEPROM que se encargará de almacenar las direcciones I2C de nuestro dispositivo (sección 4.5.1) además del *metadata* que identificará la cámara. Las hojas de características de los componentes pueden encontrarse en los siguientes anexos:

- Condensadores, Anexo B.10, B.12 y B.11.
- Bobinas, Anexo B.6.
- Resistencias, Anexo B.13.
- EEPROM, Anexo B.3.
- Diodos Zener, Anexo B.14
- Conector sensor OV5680, Anexo B.5
- Conector Molex de 30 pines, Anexo B.1

4.5.3. Módulos puente diseñados

Con respecto a los módulos puente, se han diseñado diferentes versiones del mismo preparadas para diferentes sensores de cámara. Todos los módulos puente se han desarrollado para soportar un único sensor de cámara para poder analizar la cantidad de componentes requerida por cada módulo de cámara. Las distintas versiones del módulo puente se muestran en las siguientes secciones.

Módulo puente para sensor OV5680

El funcionamiento del módulo puente, como se comenta en el capítulo 3, se basa en el paso de las líneas desde el módulo de cámara hasta la NJTX2. Además para su correcta detección por parte de la NJTX2 dispone de una EEPROM para su identificación dentro del bus i^2c como se comenta en la sección 4.5.1. Los componentes utilizados para el diseño de este módulo son los mostrados en la Tabla 4.4.

| Grupo | Componente | Valor | Encapsulado | Unidades |
|---------------|------------------|----------------------|------------------------------------|----------|
| Condensadores | C1206C104J5RACTU | 100nF | 1206 (in) | 1 |
| Resistencias | CRG1206F4K7 | 4.7kΩF | 1206 (in) | 3 |
| Conecotor | Molex | Serie FFC/FPC 30 PIN | SMT | 1 |
| Conecotor | SAMTEC | QTH-060-01-L-D-A | SMT | 1 |
| EEPROM | AT24C02D-SSHM-T | EEPROM 1.7V | TSSOP-8 4.4x3mm Pitch 0.65mm | 1 |

Tabla 4.4: Componentes del módulo puente para el sensor OV5680

- Condensadores, Anexo B.10, B.12 y B.11.
- Conecotor SAMTEC QTH-060-01-L-D-A, Anexo B.7
- Resistencias, Anexo B.13.
- EEPROM, Anexo B.3.
- Conecotor Molex de 30 pines, Anexo B.1

Como se muestra en la Tabla 4.4, se ha utilizado el coneccotor *SAMTEC QTH-060-01-L-D-A* para conectar directamente este módulo a la NJTX2 por medio del coneccotor J22 de la placa. Este coneccotor tiene 120 pines con una separación de 0.5 mm entre ellos lo cuál puede ser difícil de integrar en la fabricación teniendo en cuenta el equipo con el que contamos para ello. Por esta razón, se ha decidido diseñar otros módulos puente para sensores diferentes que simplifiquen los requerimientos de fabricación.

Módulo puente para sensor OV5647/IMX219

Este módulo tendrá el mismo funcionamiento que el expuesto en el Capítulo 3. Para su correcta detección en el bus i^2c dispone de una EEPROM, como se comenta en la sección 4.5.1. La única diferencia entre este módulo y el desarrollado para sensor OV5680, es que este ha sido diseñado para la conexión de un sensor OV5647 o un sensor IMX219 (ya que ambos tienen el mismo *pinout*).

Los componentes utilizados para el diseño de este módulo son los mostrados en la Tabla 4.5

| Grupo | Componente | Valor | Encapsulado | Unidades |
|---------------|------------------|----------------------|------------------------------------|----------|
| Condensadores | C1206C104J5RACTU | 100nF | 1206 (in) | 1 |
| Resistencias | CRG1206F4K7 | 4.7kΩF | 1206 (in) | 3 |
| Conektor | Molex | Serie FFC/FPC 30 PIN | SMT | 1 |
| Conektor | SAMTEC | QTH-060-01-L-D-A | SMT | 1 |
| EEPROM | AT24C02D-SSHM-T | EEPROM 1.7V | TSSOP-8 4.4x3mm Pitch 0.65mm | 1 |

Tabla 4.5: Componentes del módulo puente para el sensor OV5647

Como se puede ver, los componentes son los mismos que para el diseño del módulo puente para el sensor OV5680. Sin embargo, la diferencia con el anterior módulo son las interconexiones internas de la placa (debido a que el sensor OV5680 y el OV5647(Anexo E.3)/IMX219(Anexo E.5) tienen un *pinout* distinto).

Módulo puente con conexión Dupont

La última versión del módulo puente que se ha diseñado ha sido ideada para facilitar la conexión de módulos de cámara al módulo de depuración por medio de cables tipo *Dupont* hasta pines *Harwin* de conexión placa a placa rectos. Los componentes utilizados en el diseño de este módulo son los mostrados en la Tabla 4.6.

| Grupo | Componente | Valor | Encapsulado | Unidades |
|---------------|------------------|----------------------|------------------------------------|----------|
| Condensadores | C1206C104J5RACTU | 100nF | 1206 (in) | 1 |
| Resistencias | CRG1206F4K7 | 4.7kΩF | 1206 (in) | 3 |
| Conektor | Molex | Serie FFC/FPC 30 PIN | SMT | 1 |
| Conektor | HARWIN | Recto 1 fila | THD | 1 |
| EEPROM | AT24C02D-SSHM-T | EEPROM 1.7V | TSSOP-8 4.4x3mm Pitch 0.65mm | 1 |

Tabla 4.6: Componentes del módulo puente para el sensor OV5647/IMX219 con conexión DUPONT

Además de este módulo, se ha desarrollado un módulo puente similar a este mismo diseño sin la incorporación de la EEPROM para comprobar su identificación en el bus i^2c . Las hojas de características de los componentes incorporados en este diseño pueden encontrarse a continuación:

- Condensadores, Anexo B.11.

- Conector HARWIN, Anexo B.15
- Resistencias, Anexo B.13.
- EEPROM, Anexo B.3.
- OV5647, Anexo E.3
- IMX219, Anexo E.5
- Conector Molex de 30 pines, Anexo B.1

Módulo de depuración

El último módulo que ha sido diseñado, se ha denominado **módulo de depuración**. Este módulo servirá para poder conectar un dispositivo de medida a sus pines para realizar mediciones de las salidas del conector J22 de la NJTX2. Para desarrollar este módulo, han sido necesarios los componentes que se muestran en la Tabla 4.7.

| Grupo | Componente | Valor | Encapsulado | Unidades |
|----------|------------|------------------|-------------|----------|
| Conector | SAMTEC | QTH-060-01-L-D-A | SMT | 1 |
| Conector | HARWIN | Recto 1 fila | THD | 1 |

Tabla 4.7: Componentes del módulo de depuración de líneas.

En este módulo solamente se han usado los componentes que servirán de interfaz de conexión. No se han introducido más componentes debido a que pretenden medir las señales lo más directamente posible desde la NJTX2.

4.5.4. Esquemas eléctricos diseñados

Para el presente proyecto, se ha empleado una jerarquía simple de diseño, ya que los esquemas eléctricos de los circuitos completos no son lo suficientemente grandes como para usar varias hojas y se han distribuido en hojas simples. Por esta razón, se han subdividido los espacios dentro de cada una de ellas para la representación de los grupos de componentes con la identificación de su cometido dentro del diseño.

Por otro lado, el diseño del esquema eléctrico de los circuitos se presenta en varias hojas, que corresponden a cada uno de los módulos diseñados. Puede encontrar todos los esquemas eléctricos desarrollados en el Anexo F. Además puede descargar los ficheros fuente de diseño desde el repositorio en el que se almacena este proyecto [29]. Los esquemas presentes en este Anexo son:

1. **Diseño para el sensor de cámara OV5680:** Diseño de adaptación del sensor de cámara OV5680.
2. **Módulo de depuración de líneas:** Módulo de depuración de líneas del conector J22 de la NJTX2.
3. **Módulo puente sensor OV5680:** Módulo puente para la conexión del módulo de cámara con sensor OV5680 NJTX2.
4. **Módulo puente para el sensor IMX219/OV5647:** Módulo puente para los sensores IMX219 y OV5647 de *Raspberry Pi*.
5. **Módulo puente con conexión DUPONT:** Módulo puente con conexión a pines DUPONT para los sensores IMX219 y OV5647.

4.5.5. Diseño de circuitos impresos

Antes de empezar el diseño PCB de los circuitos, fue necesaria la elaboración de una librería de huellas que contuviera los conectores que usaríamos en el diseño, principalmente el *SAMTEC QTH-060-01-L-D-A* y el conector *Molex* de 30 pines. Además se han tenido que diseñar los encapsulados que no existiesen en las librerías por defecto que instala *KiCad*. Todos las huellas que se han diseñado, se han realizado siguiendo las especificaciones de las hojas de datos del Anexo B.

Por otro lado, las características técnicas comunes que definen las placas de circuito impreso son las siguientes:

- Las placas serán de tipo bicapa o de doble cara, refiriéndose al número de láminas de cobre que permitirán el trazado de pistas conductoras, que en este caso se trata de las dos caras externas de la placa: la superior o *top* y la inferior o *bottom*.
- Tanto los componentes de tecnología de agujeros pasantes (THD) como los de montaje superficial (SMD) son colocados en la capa *top* de la placa.
- Debido a la gran cantidad de componentes que alojan las placas y las restricciones dimensionales con las que se cuentan, es preciso el uso de vías para llevar a cabo las conexiones en los diferentes puntos de un mismo nodo.
- Existe la necesidad de definir un plano de tierra para señales analógicas que coincidirá con el plano de tierra de las señales digitales.

Los componentes se han colocado en cada una de las placas de tal forma que se pueda optimizar la conexión con los elementos cercanos. Además, el emplazamiento de los componentes referidos a la parte del sensor de cámaras no siguió

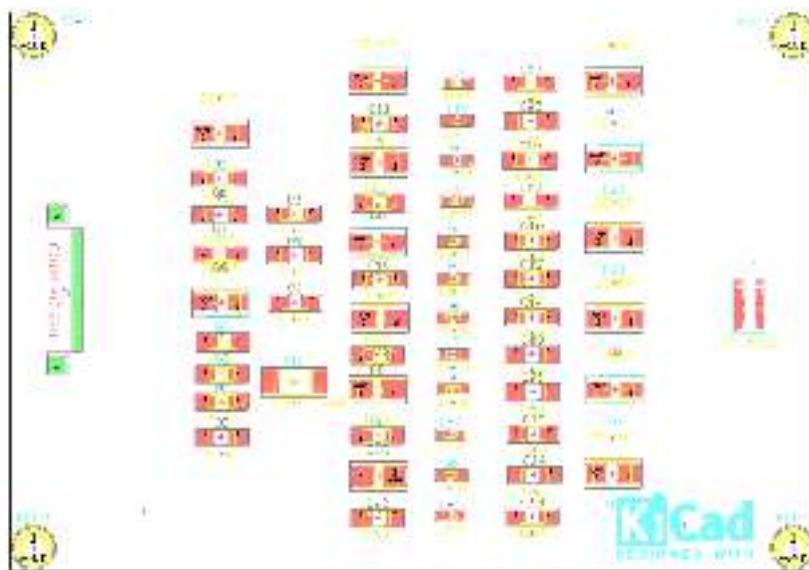


Figura 4.9: Distribución de los componentes del módulo de cámara para OV5680

ningún tipo de criterio, salvo el de intentar conseguir el trazado más corto entre elementos y mantener las distancias entre los pares de las líneas CSI-2. En las Figuras 4.9 y 4.10 pueden observarse dos ejemplos del resultado final de la colocación de los componentes en el módulo de cámara y en el módulo puente para el sensor OV5680.

Para el trazado manual de las pistas se utilizaron los valores mínimos que se indican en la Tabla 4.8. Durante el trazado de las pistas se trató aumentar al máximo permitido el ancho de las pistas y vías en las zonas donde fuera posible para así hacer más fácil la fabricación y evitar la inducción de ruido sobre las pistas. Pueden verse dos ejemplos, del resultado final del ruteado de las placas en las imágenes de las Figuras 4.11 y 4.12, donde las pistas rojas se encuentran en la cara superior de cobre y las verdes, en la cara inferior.

| | |
|--|-------|
| Margen de separación ² (mm) | 0,127 |
| Ancho de pista (mm) | 0,254 |
| Diámetro de vía (mm) | 0,889 |
| Diámetro de taladro de vía (mm) | 0,635 |

Tabla 4.8: Valores mínimos para el trazado de pistas

³El margen de separación hace referencia a la distancia mínima que debe existir entre los diferentes elementos que aloja la placa: pistas, vías y componentes.

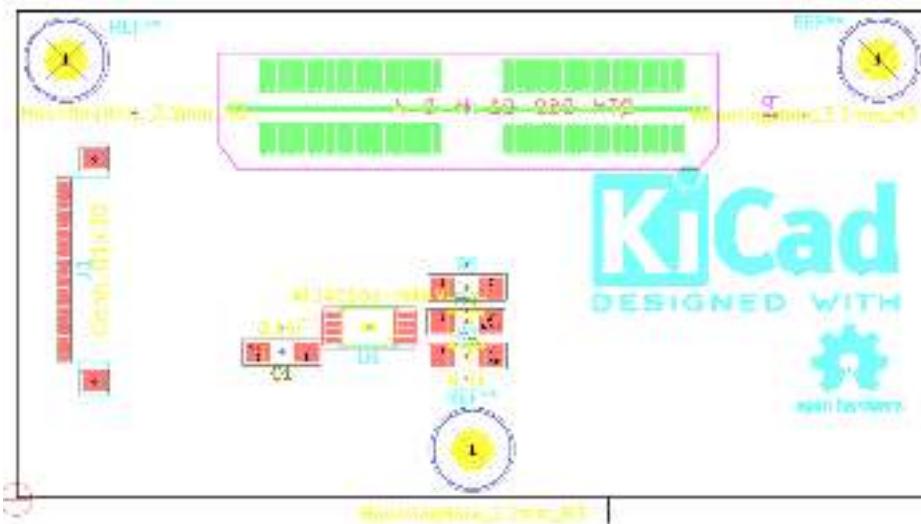


Figura 4.10: Distribución de los componentes del módulo puente para sensor OV5680

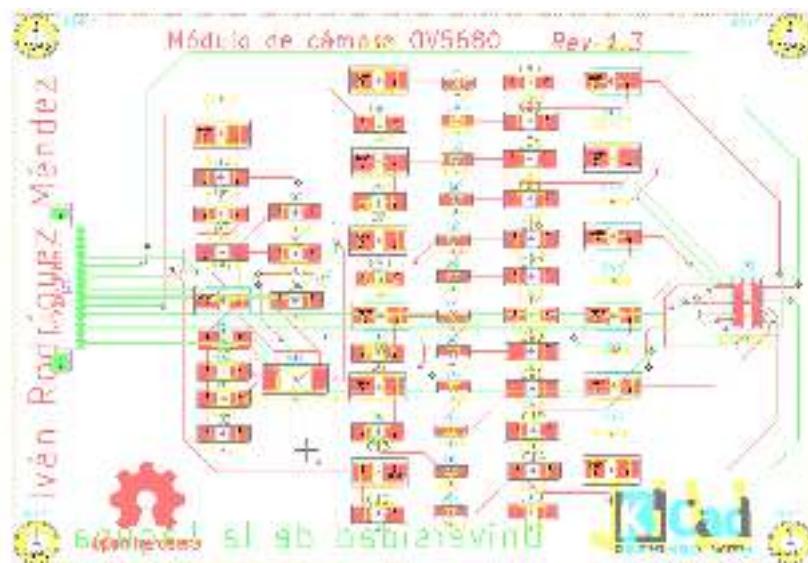


Figura 4.11: Trazado de pistas en el diseño PCB del módulo de cámara para el sensor OV5680

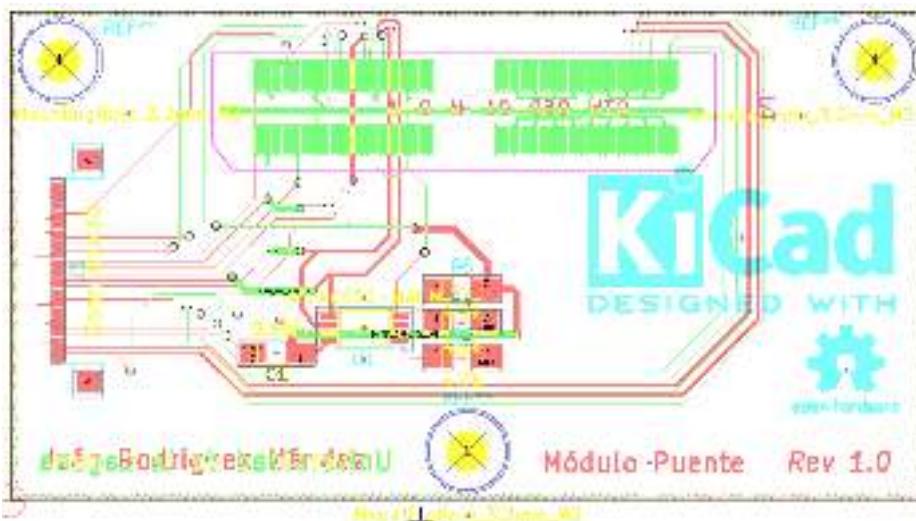


Figura 4.12: Trazado de pistas en el diseño PCB del módulo puente para sensor OV5680

Aunque se usan señales analógicas y digitales en las placas, se define un único plano de tierra para ambas. Así, en la cara *top* se define un plano de tierra y se conecta con el de la cara *bottom*, por medio de una vía. En ambos casos, los planos de tierra se conectan con las respectivas conexiones de tierra (GND y AGND) mediante nodos térmicos⁴, donde las pistas de dichas señales se integran en el plano, es decir, no se distinguen mediante un contorno de aislamiento, como sucede con el resto de conexiones.

Con el fin de acoplar la placa a la NJTX2, en el caso del módulo puente y el de depuración, se han añadido los agujeros mecánicos de sujeción de 3.2 mm de diámetro en los puntos donde fuera necesario, con la única finalidad de alojar tornillos de esas dimensiones. Dichos taladros son equidistantes entre sí y están situados a una distancia mínima de 1mm con respecto al lugar previsto para cualquier pista o componente, con el fin de evitar cualquier tipo de daño durante el mecanizado de la placa. Además, también se ha delimitado el contorno de la placa a una distancia de 3mm respecto a los cuatro agujeros de sujeción.

Para distinguir los módulos entre sí, se ha añadido el nombre de cada uno de ellos en la capa frontal de cobre. Por ejemplo, si se trata del módulo puente la inscripción será (“Módulo puente”). Por otro lado, también se indica la versión de la placa, el nombre del diseñador y se ha colocado el logotipo oficial de la Universidad de La Laguna. Esta información añadida aparecerá en la capa de cobre ya que así se podrá integrar dentro del proceso de fabricación.

⁴Los nodos térmicos son pads específicos que permiten la conexión de una determinada señal con su correspondiente plano de cobre, además de conseguir una mejor disipación de calor.

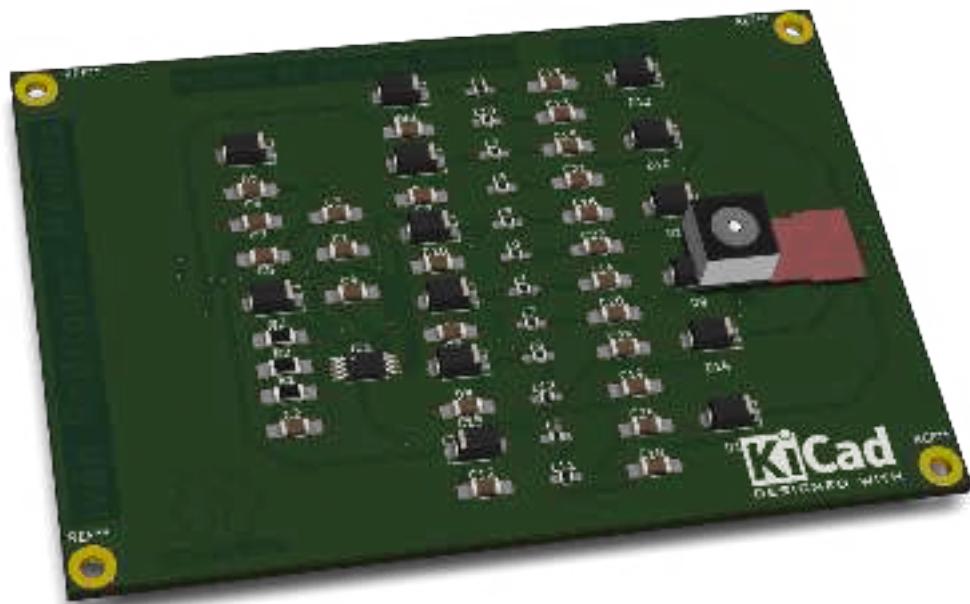


Figura 4.13: Visualización 3D del módulo de cámara para el sensor OV5680

En las placas donde así se requiera, se utilizará la colocación de texto en la placa de cobre para indicar el nombre de los pines de los componentes. En el caso del módulo puente esto cobra especial importancia ya que en su diseño se han indicado el nombre de los 87 pines de los que dispone el módulo.

Por último, para la fabricación de la PCBs diseñadas es preciso generar los documentos y archivos de post-proceso que permitirán su construcción y montaje. Así, los archivos GERBER generados corresponden a las capas de cobre (*top* y *bottom*), a la máscara de soldadura y a la serigrafía, que pueden visualizarse en el anexo G en forma de fotolitos. También pueden encontrarse los ficheros fuente en el repositorio donde se almacena el proyecto [29].

4.5.6. Modelos 3D de las placas diseñadas

Con el fin de tener una aproximación del aspecto de las placas sin estas haber sido fabricadas, se han desarrollado sus modelos 3D. Para ello, se han utilizado componentes ya existentes en la red [21], por lo que sólo ha sido necesario ubicarlos correctamente en las PCBs. En dichos modelos se podrán apreciar los elementos que alojan las placas, que consisten en los componentes del circuito, las pistas que los conectan y sus vías correspondientes. En las Figuras 4.13 y H.3 se pueden ver algunos modelos 3D de los que se han desarrollado. Para ver todos los modelos 3D diseñados véase el Anexo H.



Figura 4.14: Visualización 3D del módulo puente para el sensor OV5680

4.6. Viabilidad de diseños preliminares

Una vez realizados todos los diseños preliminares, se evalúan para comprobar la viabilidad de su fabricación. Como se ha visto en secciones anteriores, el utilizar un sensor como el OV5680 obliga al desarrollo y fabricación de un módulo de cámara para la adaptación de dicho sensor. Por contra, si se eligen sensores como el IMX219 y el OV5647, estos vendrán con la electrónica necesaria para su funcionamiento incorporada. Por esta razón, se ha tomado la decisión de adquirir **módulos de cámara** con sensores IMX219 (ya que las características de los sensores IMX219 son mejores que las de los sensores OV5647).

Por otro lado, con respecto al desarrollo de módulos puente, se han mostrado varios desarrollos diferentes, aunque todos enfocados para el uso de un solo sensor. Como se ha comentado, este proyecto se basa en la incorporación de 6 módulos de cámara a la NJTX2 por lo que se requiere una alta capacidad de fabricación para llegar a integrar la electrónica suficiente como para lograr integrar 6 conexiones para módulos de cámara en un módulo puente con su correspondiente electrónica en las dimensiones establecidas por Nvidia. Por ello, gracias a la vigilancia tecnológica durante el desarrollo del proyecto, se ha decidido adquirir el **módulo puente** denominado **J20** de la empresa Auvidea. Este módulo es capaz de incorporar hasta 6 módulos de cámara con sensor IMX219/OV5647 por lo que se adaptaría a los requerimientos del desarrollo del proyecto en cuanto al uso de sensores de cámara de bajo coste. Para más información véase el Anexo C.7.

Por último, con respecto al **módulo de depuración**, se ha decidido que su fabricación puede aportar mucha información al proyecto debido a que permitirá el análisis de las señales salientes de la NJTX2. Esto resulta de gran utilidad a la hora de desarrollar el software y los *device tree* y los módulos del *kernel*.

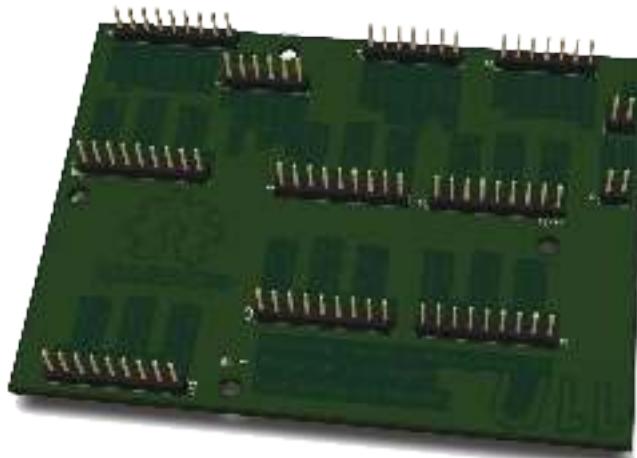


Figura 4.15: Visualización 3D del módulo de depuración

4.7. Módulos a implementar en el desarrollo

Como se ha visto en la evaluación de la viabilidad de los módulos, se han seleccionado aquellos que se han decidido fabricar y aquellos, que por contra se comprarán. Finalmente los módulos seleccionados para incluirlos en el desarrollo del proyecto serán:

- **Módulo de cámara:** Para los módulos de cámara se han seleccionado los módulos con sensor IMX219 desarrollados para Raspberry Pi, concretamente para la Cámara de Raspberry Pi versión 2.1. Se muestran en la Figura 4.8c
- **Modulo puente:** Para el módulo puente se ha decidido adquirir el módulo J20 desarrollado por Auvidea. Puede verse este módulo en la Figura 4.16
- **Módulo de depuración:** Se ha decidido fabricar el módulo de depuración por su gran utilidad a la hora de analizar las señales de la NJTX2. Puede verse un modelo 3D de este módulo en la Figura 4.15.



Figura 4.16: Módulo J20

Capítulo 5

Fabricación de PCBs

En este capítulo se introducen los aspectos y pasos a tener en cuenta a la hora de fabricar las placas de circuito impreso del presente proyecto. Como se comenta en la sección 4.7, el módulo a fabricar será el denominado **módulo de depuración**. Por lo tanto, durante este capítulo, se verán los aspectos generales a tener en cuenta en la fabricación y concretamente los aspectos relacionados con la fabricación del módulo de depuración.

5.1. Características físicas de las placas

En el capítulo 4 se describieron las características técnicas que deben tener las placas a la hora de diseñarlas. Sin embargo, también es preciso definir las características físicas que estas han de tener para poder ser fabricadas. Además de la información que aportan los archivos GERBER, también debemos tener en cuenta el tipo de placas de cobre con las que se pretende realizar la fabricación y sus características:

- Grosor de la PCB: el espesor del sustrato de la tarjeta debe ser de 1,6mm¹.
- Grosor de las láminas conductoras: el espesor del cobre debe ser de $35\mu\text{m}$ ¹.
- Color de la máscara: el color de la máscara de soldadura ha de ser verde (en el caso de aplicarla). En su defecto se aplica laca transparente para proteger la placa.

¹Estos valores de espesores corresponden a los típicos que se emplean en placa de doble cara.

5.2. Metodología de fabricación de PCBs

En esta sección se explicará la metodología seguida para la fabricación de placas de circuito impreso en este proyecto.

Fabricación de placas de circuito impreso con CNC

Para la fabricación de placas de circuito impreso con máquinas de control numérico como las mostradas en las imágenes de las Figuras 2.1 y 2.2, es necesario generar algunos ficheros de diseño en la fase de *diseño del circuito impreso* mostrada en el capítulo 4. Esos ficheros son los siguientes:

- Ficheros Gerber(Extensión .gbr): A la hora de trazar el diseño se deben los ficheros gerber de las capas que se quieren fabricar por medio de una CNC. Típicamente las capas suelen la frontal y trasera de cobre ya que son las que la máquina se encargará de fresar. Por otro lado, también se pueden generar ficheros gerber para colocar la pasta de soldadura en nuestro circuito impreso (en el caso de que nuestra máquina de control numérico estuviera preparada para realizar dicha tarea)
- Fichero de taladros(Extensión .drl): Se deben generar estos ficheros si se desea que la máquina realice los taladros de nuestro circuito impreso (si los tuviera) además de los agujeros de montaje.
- Fichero de posiciones de huellas(Extensión .pos): Si se dispone de una máquina capaz de posicionar los componentes automáticamente, se puede generar un fichero que contenga la posición de cada una de las huellas para que la máquina coloque los componentes.

Una vez generados los ficheros necesarios se puede pasar a generar los ficheros de trayectorias de máquina también conocidos como ficheros G-Code. Para generar dichos ficheros, se deben conocer las características de la máquina que se utilizará (límites de los ejes, velocidades y aceleraciones máximas,etc) para fabricar el circuito. Por otro lado, también se debe tener bien caracterizada la herramienta que se utilizará para cada uno de los pasos de fabricación. Las operaciones básicas a realizar por la CNC son las siguientes:

- *Milling*: Operación de eliminación de cobre de la placa de circuito impreso. Esta es la operación más estándar dentro del trabajo con una CNC.
- *Drilling*: Operación de realización de taladros. Operación de taladrado de la placa de circuito impreso. Suele aplicarse después de la operación de *milling*.

- *Cutting*: Operación de corte de la placa. Esta operación se realiza para ejecutar el corte de los bordes de la placa de circuito impreso. Suele ser una operación aplicada al final del proceso de fabricación.

Teniendo en cuenta las operaciones básicas a realizar con la CNC, se usará *FlatCAM* para pasar los ficheros con extensiones **.gbr** y **.drl** a ficheros con extensión **.gcode**. En este programa se indicará cuál es el tamaño de la herramienta que se usará para realizar las operaciones enumeradas anteriormente. Posteriormente se genera un fichero para cada una de las operaciones a ejecutar. Para comprobar las trayectorias generadas por *FlatCAM* en el G-code se puede utilizar *CAMotics* para la visualización de las trayectorias.

Una vez comprobados los G-Code, se cargarán en la máquina dichos ficheros. Previamente se debe colocar en la CNC una placa de cobre que tenga el número de caras requerido (doble cara o simple) y que tenga dimensiones suficientes como para contener la PCB que se fabricará. Una vez hecho esto, se ejecutarán los Gcode que se han generado, haciendo en primer lugar las operaciones de fresado, seguidamente las operaciones de taladrado y por último, las operaciones de corte.

Después de realizar todas las operaciones con la CNC, puede procederse a la colocación de los componentes en la PCB y su soldadura.

Fabricación de placas de circuito impreso con químicos

Para la fabricación de placas de circuito impreso con procesos químicos, es necesaria la generación de algunos ficheros en la fase de *diseño del circuito impreso* (Capítulo 4), tal y como ocurría para la fabricación con máquinas de control numérico. En este caso, los ficheros que se deben generar deberán estar en formato **.pdf** o preferiblemente en **.svg** para evitar los cambios de escala. Los ficheros que se deben generar serán aquellos que se quieran transferir al circuito, comúnmente serán las capas frontal y trasera de la PCB. En el caso de la capa trasera, esta debe estar generada en trazado invertido o espejo, ya que tanto si se hace por medio de la técnica de isolado como con la técnica de transferencia de toner (como será el caso del presente proyecto) esta quedaría transferida del revés. Una vez generados los ficheros, se usa *inkscape* para imprimir dichos ficheros asegurando que no hay un cambio en las dimensiones del diseño.

Para la impresión, se utiliza un papel con una película de cera que mejora la transferencia del toner a la placa de cobre virgen. La impresora a utilizar debe ser una impresora láser. Una vez impresos los fotolitos de la capa frontal y trasera, hay que recortarlos y alinearlos con ayuda de algún punto de referencia dentro del diseño, normalmente será un agujero de alineación. Posteriormente, se coloca la placa de cobre entre ambos fotolitos y se le aplica calor con ayuda de una plastificadora de rodillos (también podría utilizarse para este cometido una plan-

cha). Después de unos 6 minutos de aplicar calor y presión a los fotolitos se puede extraer la placa de la plastificadora y retirar el papel encerado. Si todo ha salido correctamente, la tinta habrá quedado adherida al cobre.

El siguiente paso a realizar, será el ataque químico de la placa. Este procedimiento se lleva a cabo para eliminar el cobre que no ha quedado cubierto por la transferencia del fotolito. Para dicho fin se usará **Persulfato de sodio**. Preparando una mezcla adecuada en una cubeta de ácido con agitación a 40 °C, se debe introducir la placa de 6 a 10 minutos en función del número de usos previos que haya tenido dicha mezcla para fabricar otras placas. Transcurrido este tiempo, se observará que el cobre sobrante a desaparecido, quedando únicamente el toner que se había transferido a la placa.

Después realizar el ataque químico, se debe eliminar el toner que ha quedado en la placa para que quede en esta únicamente el cobre que formara parte del diseño. Para eliminarlo, se usa acetona aplicada sobre papel y frotando este contra la placa. Es posible que se tenga que repetir el proceso varias veces, cogiendo un nuevo trozo de papel en cada una. Una vez se ha eliminado el toner, se puede pasar a realizar los agujeros sobre la placa.

Para la realización de los agujeros (en el caso de que el diseño los tenga), se puede usar una *dremel* o un taladro de banco. Si en la etapa de generación de los ficheros de fabricación se ha generado el fichero de taladros, se dispondrá de una referencia del diámetro de cada uno de los agujeros dentro de la placa. Una vez conocido el diámetro de los agujeros y en disposición de las diferentes brocas, se puede proceder a realizarlos teniendo en cuenta que para evitar la rotura de las brocas la velocidad de avance debe ser baja y la de giro moderada.

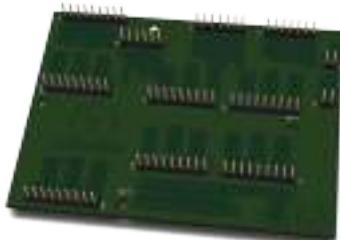
Por último, una vez realizados los agujeros, se pasa al proceso de soldadura.

Soldadura de placas de circuito impreso

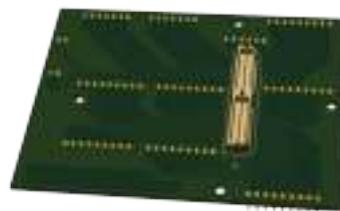
Una vez fabricada la placa de circuito impreso, ya sea por el método clásico del ataque químico o por medio de una máquina de control numérico, se debe realizar la soldadura de los componentes. Antes de todo ello, hay que limpiar la placa de cobre con acetona o con alcohol isopropílico para evitar la presencia de restos del proceso de taladrado o suciedad en la placa.

Para la soldadura, existen principalmente dos técnicas: la soldadura con soldador manual y la soldadura con horno. Típicamente la soldadura manual se realiza con componentes THT (*Through-Hole Technology*) y la soldadura con horno se aplica a componentes SMT (*Surface Mount Technology*). se explican estas dos técnicas por separado para una mayor claridad:

- **Soldadura manual:** Para la soldadura manual, se deben colocar los componentes en la placa en primer lugar. Si estos componentes son THT, se



(a) Módulo de depuración cara superior.



(b) Módulo de depuración cara inferior.

Figura 5.1: Visualización 3D de ambas caras del módulo de depuración.

pasarán a través de los agujeros para sujetarlos. Una vez colocados todos los componentes, se pondrá el estaño en contacto con una de las patas del componente y el soldador (a una temperatura aproximada de 300 °C) en contacto con la misma pata. Por efecto de la temperatura, el estaño pasará a estado líquido y se adherirá a la pata del componente. En ese momento, se retirará el soldador haciendo que el estaño se enfrié y el componente quede soldado. Es importante no tocar con la punta del soldador directamente el estaño ya que es muy probable que se formen bolas o soldaduras frías. Por otra parte, se puede utilizar *flux* para mejorar la soldadura.

- **Soldadura con horno:** Para la soldadura con horno, se deberá utilizar lo que se conoce como pasta de soldadura. Esta pasta, es una mezcla de estaño y otros compuestos que hacen que sea algo líquida y facilite su aplicación. Para dicha aplicación, se utilizarán diferentes aplicadores (de diferentes tamaños) para aplicar la pasta de soldadura en los pads de la placa de circuito impreso. Como ya ha comentado, esta técnica de soldadura se aplica en componentes SMT. Para realizarla, se aplica la pasta de soldadura sobre los pads del circuito y posteriormente se colocan los componentes en su lugar. Es importante no aplicar demasiada pasta de soldadura ya que si se hace es probable que el componente no se suelde correctamente. Después de colocar la pasta en todos los pads y los componentes en su lugar, se introducirá la placa en el horno. Una vez introducida, hay que seleccionar la curva de temperatura que se estime oportuna para soldar los componentes que se han colocado y confirmar la selección. Después de 6 o 7 minutos, cuando haya terminado el proceso de soldadura, mediante una señal sonora el horno indicará que el proceso ha finalizado.



Figura 5.2: Placas de cobre de doble cara

5.3. Fabricación del módulo de depuración

Para la fabricación del módulo de depuración mostrado en la Figura 5.1, se han seguido los pasos detallados en la sección 5.2. En las siguientes secciones se documenta el procedimiento seguido acompañado de imágenes de cada etapa.

5.3.1. Proceso de fabricación de la PCB

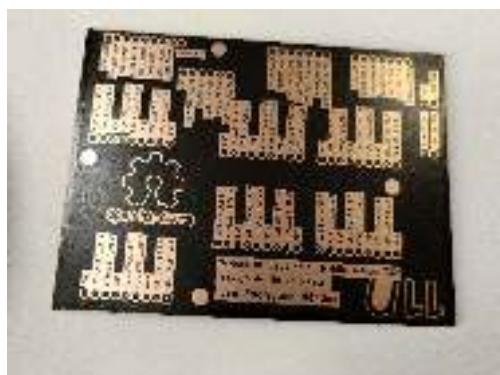
La fabricación de la placa de circuito impreso se ha realizado usando el procedimiento de atacado químico debido a que la separación entre las patas del conector *SAMTEC QTH-060-01-L-D-A*, usado para este módulo, tiene una separación de 0.2 mm entre sus patas. La fabricación de una PCB con estas separaciones mínimas entre pistas mediante una CNC puede resultar un proceso bastante complejo además de requerir una gran calibración para la máquina. Por esta razón, el módulo de depuración se ha fabricado mediante el atacado químico.

El primer elemento necesario para la fabricación del módulo de depuración es una placa de cobre virgen como las que se muestran en la imagen de la Figura 5.2. No es necesario que las placas de cobre tengan ninguna característica especial, como por ejemplo, fotosensibilidad. Por otro lado, son necesarios los ficheros de fabricación comentados en la sección 5.2, en el caso del módulo de depuración se usan los fotolitos de la cara frontal y trasera. Una vez impresos los fotolitos sobre el papel encerado, se transfieren por medio de una plastificador de rodillos como la de la imagen de la Figura 5.3.

Debido al tamaño del módulo de la PCB, es posible tener ciertas dificultades para pasar la PCB por los rodillos, por ello lo recomendable es introducirla con el lado más largo paralelo a los rodillos. Una vez terminado el proceso de transfe-



Figura 5.3: Proceso de transferencia de fotolito por plastificadora de rodillos



(a) Transferencia de toner cara superior.

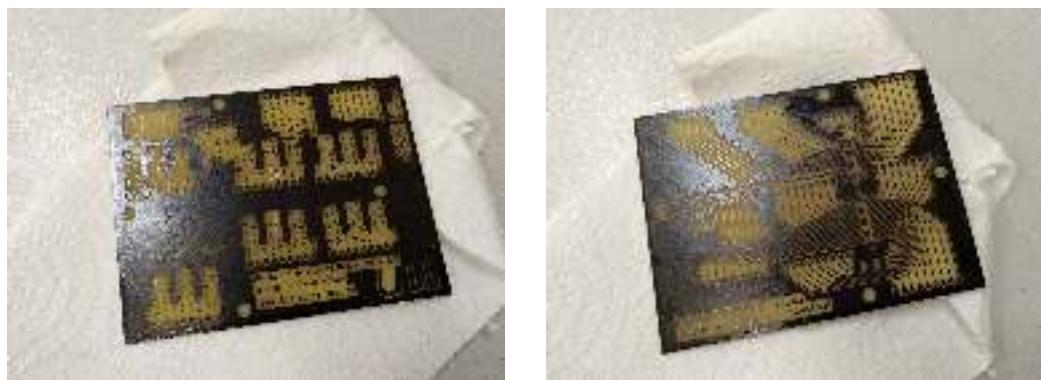


(b) Transferencia de toner cara inferior.

Figura 5.4: Resultado de la transferencia del toner sobre la placa de cobre.



Figura 5.5: Ataque químico a la placa de cobre



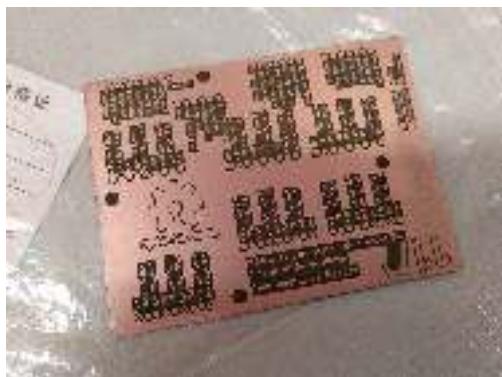
(a) Resultado ataque químico de cara superior. (b) Resultado ataque químico de cara inferior.

Figura 5.6: Resultado del ataque químico la placa de cobre.

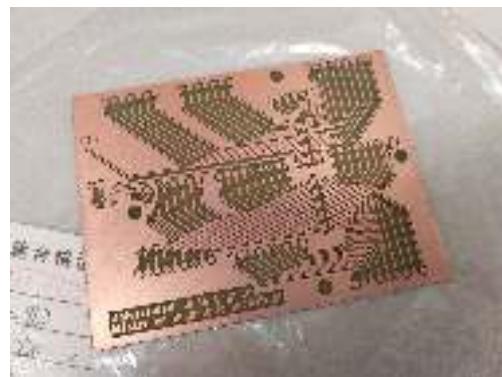
rencia de los fotolitos, las placas tendrán el aspecto de las imágenes mostradas en la Figura 5.4.

Una vez realizada la transferencia del toner en las placas, se prepara la mezcla de **Persulfato de sodio** con la que se realizará el atacado. Para un mayor control sobre el atacado de la placa se introduce la mezcla en una cubeta en la que también introduciremos la placa, tal y como se muestra en la imagen de la Figura 5.5. El resultado una vez finalizado el ataque con el ácido es el mostrado en las imágenes de la Figura 5.6.

Llegado este punto, se puede aplicar acetona sobre la placa de circuito impreso para eliminar el toner que ha quedado adherido sobre el cobre. Para la aplicación de la acetona es recomendable usar papel absorbente y guantes para evitar el contacto con la piel. Una vez realizada la limpieza del toner aplicado sobre la placa el resultado es el mostrado en la imagen de la Figura 5.7.



(a) Resultado de limpieza con acetona de la cara superior.



(b) Resultado de limpieza con acetona de la cara inferior.

Figura 5.7: Resultado de la limpieza con acetona de la PCB.



Figura 5.8: Corte de pistas con bisturí

Cuando se han retirado todos los posibles restos de toner de la PCB se pueden observar los posibles defectos que esta tenga debido a los fallos que se hayan podido producir en el proceso de fabricación. Típicamente, el fallo más común es la unión de pistas debido a un tiempo de ataque químico insuficiente o por un defecto en la impresión del fotolito o en su transferencia. Para corregir estos errores en este punto de la fabricación, se puede utilizar un bisturí y una lupa, en nuestro caso electrónica. El bisturí sirve para realizar manualmente un corte entre las dos pistas que se pretende separar. Puede verse un ejemplo en la imagen de la Figura 5.8.

Una vez comprobada la correcta conductividad de todas las pistas y conexiones de la placa, se realizan los agujeros con ayuda de una dremel como la de la imagen de la Figura 5.9.

Realizados los agujeros de montaje con las herramientas correspondientes, se puede pasar a la etapa de soldadura de los componentes sobre la placa.



Figura 5.9: Dremel para realizar los agujeros de los conectores



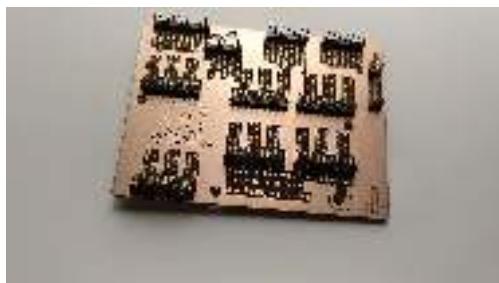
(a) Aplicación de pasta de soldadura.

(b) Resultado de la soldadura.

Figura 5.10: Proceso de soldadura con horno.

5.3.2. Soldadura de conectores

Para la soldadura de los componentes en la placa, se usa la técnica manual y la soldadura con horno, ambas vistas en la sección 5.2. El orden para la soldadura de los componentes será en primer lugar aquellos que necesiten ser soldados usando el horno y por último los que se suelden manualmente. Para la soldadura con horno se aplica la pasta de soldadura sobre los pads donde se colocarán los componentes. En el caso del módulo de depuración, el único componente que se soldará con el horno es el SAMTEC QTH-060-01-L-D-A. Una vez colocada la pasta de soldadura sobre los pads, se coloca el componente manteniendo la alineación para evitar que se suelden las patas entre sí. Finalmente se introduce la placa en el horno durante 7 minutos con una curva de temperatura gradual con un máximo a 250 °C. En la imagen de la Figura 5.10, pueden verse dos imágenes de parte del proceso. En la imagen de la Figura 5.10a se ve como se ha realizado la aplicación de la pasta de

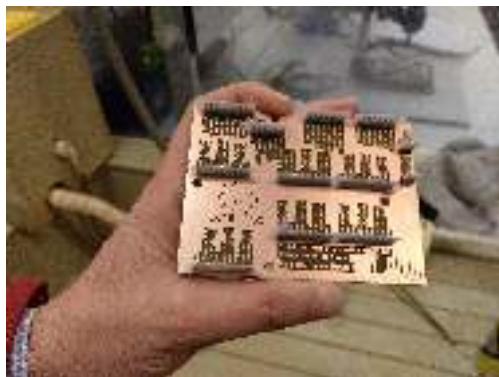


(a) Resultado de la soldadura en la cara superior.

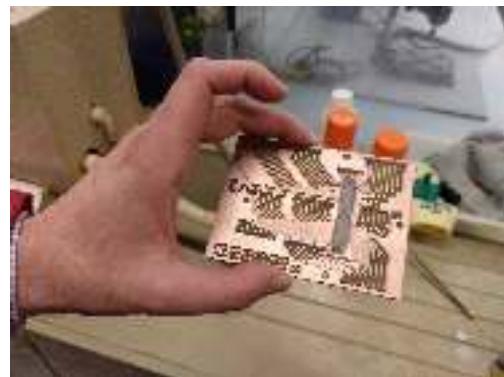


(b) Resultado de la soldadura en la cara inferior.

Figura 5.11: Resultado del proceso de soldadura.



(a) Protección para aplicar laca en la cara superior.



(b) Protección para aplicar laca en la cara inferior.

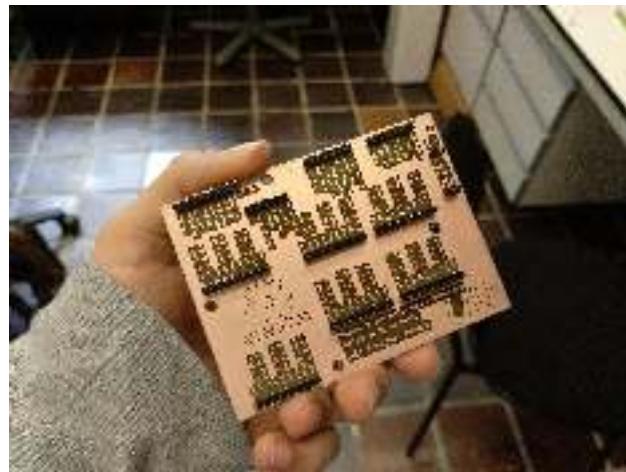
Figura 5.12: Protecciones para aplicar laca a la PCB.

soldadura. En la imagen de la Figura 5.10b se muestra el resultado final una vez soldado el conector.

Para la soldadura manual, se colocan los conectores rectos *Harwin* de conexión placa a placa en sus agujeros. Posteriormente, aplicando la técnica explicada en la sección 5.2, se soldarán a la placa. El resultado final de la soldadura de la placa de circuito impreso puede verse en la imagen de la Figura 5.11.

5.3.3. Aplicación de laca a la PCB

Para la protección del circuito impreso, es recomendable la aplicación de una película protectora. En el caso del módulo de depuración, se ha aplicado laca transparente tanto a su cara superior como a su cara inferior. Para evitar que la laca entre en contacto con los conectores, se forran usando cinta adhesiva. En la



(a) Resultado final de la fabricación del módulo de depuración. Cara superior.



(b) Resultado final de la fabricación del módulo de depuración. Cara inferior.

Figura 5.13: Resultado final de la fabricación del módulo de depuración.

Imagen de la Figura 5.12 pueden verse las protecciones. Finalmente, una vez se ha secado la laca, el proceso de fabricación de la PCB ha finalizado.

5.3.4. Resultado final

Una vez completados todos los pasos para la fabricación del módulo de depuración el resultado final es el mostrado en las imágenes de la Figura 5.13. Por otro lado, se puede ver en la imagen de la Figura 5.14 la imagen del módulo de depura-

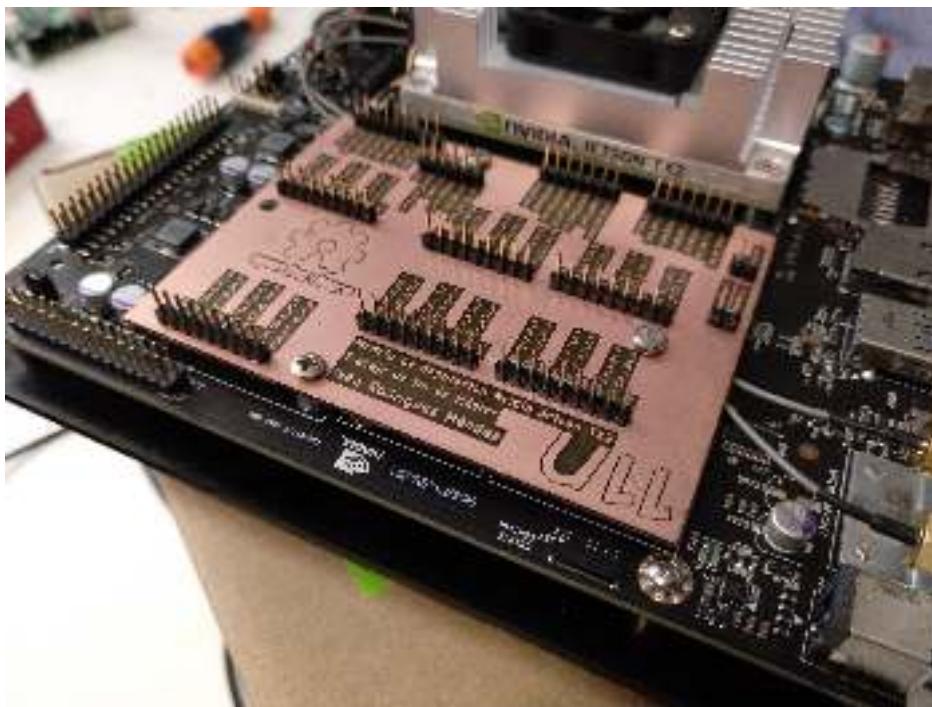


Figura 5.14: Módulo de depuración integrado en Nvidia Jetson TX2

ción integrado con la Nvidia Jetson TX2. Como puede observarse, se ha montado la placa de depuración sobre los agujeros que se encuentran en la NJTX2.

Capítulo 6

Diseño y fabricación mecánica

Como se ve en el Capítulo 3, es necesario desarrollar una estructura mecánica de soporte que de sustento a la NJTX2 además del *array* de cámaras. Por otro lado, también debe ser diseñado y fabricado un elemento mecánico que permita la calibración de las cámaras, esto quiere decir que debe ser posible variar el ángulo de apuntado de las mismas. Por ello, el objetivo es diseñar una estructura que resulte ligera a la par que resistente y pueda ser fabricada por métodos de prototipado rápido como las impresoras 3D o las cortadoras láser. En las siguientes secciones se introduce la metodología seguida, el diseño de los elementos mecánicos y su fabricación.

6.1. Diseño de la estructura mecánica

En esta sección se introducen los diferentes elementos mecánicos que se han diseñado además de la metodología seguida para su diseño.

6.1.1. Elementos principales a diseñar

Como se comenta en el Capítulo 3, en el desarrollo mecánico de este proyecto se deben diseñar dos elementos mecánicos fundamentales: el **calibrador de cámaras** y la **estructura mecánica de soporte**. Las características de ambos elementos se presentan a continuación.

Calibrador de cámaras

Como se indica en el Capítulo 4, el módulo de cámara a utilizar será el módulo de cámara de *Raspberry Pi* Versión 2.1. Para que resulte posible orientar los módulos de cámara y su ángulo de apuntado, se debe diseñar una estructura que

sea capaz de girar los módulos sobre dos de sus ejes (eje x e y). Por otro lado, el calibrador también debe servir para fijar el módulo de cámara. En la Figura 3.3 se puede ver un diagrama con los posibles giros del módulo de cámara con respecto a sus ejes. Adicionalmente, los diseños deben asegurar la posibilidad de añadir nuevos módulos de cámara al calibrador desde la fase de diseño. Esto quiere decir, que el módulo debe estar diseñado para ser duplicado y usado para 6 cámaras.

Estructura de soporte del sistema

Para el sustento de todos los elementos del sistema, se debe desarrollar una estructura simple que permita la integración y fijación del calibrador de cámaras y la NJTX2. La estructura se diseñará para ser lo más simple y ligera posible. Además debe tener la posibilidad de poder mover la posición del calibrador de cámaras en ella y por otro lado la posición de la NJTX2.

6.1.2. Metodología de diseño

En las siguientes secciones se introducen los procedimientos seguidos comúnmente para el diseño de los elementos mecánicos que integran este proyecto.

Diseño de elementos mecánicos para impresora 3D

Para el diseño de elementos mecánicos a fabricar con impresoras 3D, en primer lugar hay que realizar un boceto con las medidas de las piezas que se pretenden diseñar. Este mismo boceto, se introduce en *FreeCAD* con el banco de trabajo denominado *Part Design*. En este conjunto de ventanas, se dibujarán los planos base de la pieza y posteriormente se realizan las extrusiones. Una vez extruída la pieza, se pasa al banco de trabajo denominado *Part* donde se pueden realizar agujeros a la pieza o realizar operaciones *boolenas* con otras piezas. Estas operaciones pueden ser uniones, intersecciones, diferencias, etc.

Una vez diseñada la pieza, se podrá exportar el resultado en formato **.obj** o bien generar un plano dentro del banco de trabajo *Drawing* y *Drawing Dimensioning* el cuál forma parte de una macro que puede instalarse dentro del programa y que ayuda a realizar las cotas de los planos.

Diseño de piezas para corte láser

Otra técnica utilizada en el desarrollo de este trabajo, es el corte de piezas con una cortadora láser como la de la imagen de la Figura 2.5. Para el corte de piezas con esta máquina, los diseños deben estar en formato **.svg**, para crear dichos diseños se utiliza *inkscape*. Hay que tener en cuenta que para que la máquina sea

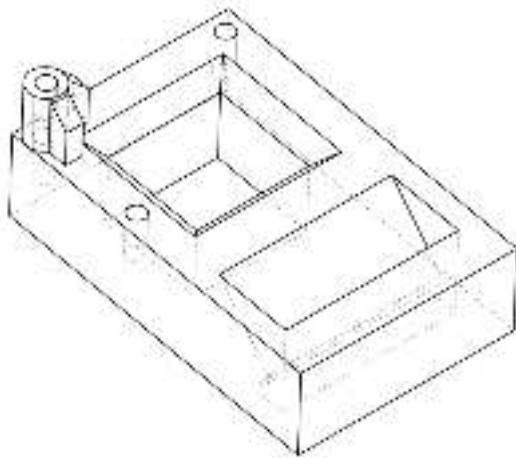


Figura 6.1: Diseño del módulo de calibración de cámaras

capaz de interpretar el fichero se debe seguir un código de colores que dependerá del trabajo que se desea que realice la máquina en dicha zona. El código de colores es el siguiente:

- **Rojo:** Este color será utilizado para aquellas zonas en las que queremos que la máquina realice operaciones de corte.
- **Azul:** Este color será utilizado para aquellas zonas donde se realizarán operaciones de grabado.
- **Negro:** Este color será utilizado para aquellas zonas donde se quiera aplicar un rebaje de la superficie.

Una vez realizado los diseños con ayuda del software mencionado, se pueden cargar los ficheros en la cortadora para proceder a su uso.

6.1.3. Elementos diseñados

Diseño del módulo de calibración de cámaras

Se puede ver el resultado para el diseño del módulo de calibración en el boceto de la Figura 6.1. Para que este módulo funcione deben usarse tornillos de 2 mm de diámetro con una longitud de al menos 12 mm. Además de los tornillos son necesarios resortes con una longitud en reposo de 8 mm y 4 mm comprimidos. Para el montaje de los módulos de cámara y su correcto funcionamiento, se insertarán los tornillos en los agujeros dispuestos para ello. En dos de los agujeros, se insertarán los tornillos por el interior de los resortes, de esta forma se tendrá un sistema de

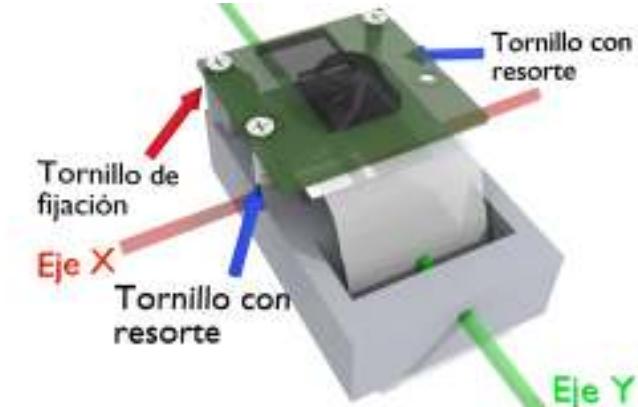


Figura 6.2: Modelo 3D de fijación de tornillos en el módulo de calibración

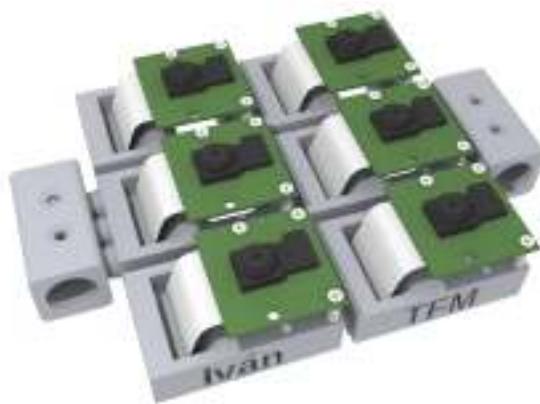


Figura 6.3: Modelo 3D del calibrador para 6 cámaras

basculación. El tercer agujero servirá para fijar el módulo al soporte. Podemos ver el montaje de los tornillos y resortes en el modelo 3D de la Figura 6.2.

Una vez realizado el montaje, el funcionamiento de la calibración es simple. Para variar la rotación de la cámara en uno de sus ejes basta con apretar o aflojar uno de los dos tornillos que disponen de resorte. Debido a la presentación del resorte, el módulo siempre permanecerá pegado a la cabeza del tornillo lo cuál permite el funcionamiento del ajuste debido a que la cabeza del tornillo bajará o subirá en función de si se aprieta o se afloja.

Para la integración de todos los módulos de cámara, se ha diseñado una estructura que une varios módulos individuales como el de la Figura 6.1. Puede verse la integración de 6 módulos de calibración en el modelo 3D de la Figura 6.3.

En el Anexo J pueden encontrarse los planos realizados en el diseño del módulo de calibración de cámaras. También pueden encontrarse los ficheros fuente en el repositorio donde se almacena el proyecto [29].



(a) Vista frontal del modelo 3D de la estructura de soporte.



(b) Vista trasera del modelo 3D de la estructura de soporte.

Figura 6.4: Modelo 3D de la estructura de soporte del sistema.

Diseño de la estructura de soporte

El resultado del diseño de la estructura mecánica puede verse en las imágenes de la Figura 6.4. En el modelo de la Figura 6.4a se puede observar el frontal de

la estructura de soporte. En esta zona se ha situado el ajustador de cámaras de la Figura 6.3. Para ello se han utilizado dos tubos travesaños que además pueden deslizarse verticalmente. Estos travesaños se sujetan con el resto de la estructura por medio de una pieza de unión con los pilares. Adicionalmente, esta misma pieza permite la fijación de los travesaños en una posición concreta gracias al uso de tornillos de 2 mm.

Por otro lado, en el modelo 3D de la Figura 6.4b puede verse la zona trasera de la estructura de soporte. En esta zona se encuentra la NJTX2 sustentada sobre una mesa sujeta por 4 pilares verticales. La mesa puede moverse gracias a que sus soportes deslizan sobre los tubos verticales. Para su fijación, al igual que para la fijación de los travesaños que soportan las cámaras, se usan tornillos de 2 mm que aprisionarán los soportes contra los tubos, evitando así su movimiento.

Por último, para que la estructura resulte ligera, los tubos utilizados en el diseño son huecos.

En el caso del diseño de la mesa de la NJTX2, se ha diseñado usando *inkscape* teniendo en cuenta las medidas de la placa. Este soporte se ha diseñado para que resulte posible la fijación de la NJTX2 sobre él por medio de tornillos y tuercas de 3.2 mm.

En el Anexo J pueden encontrarse los planos realizados en el diseño de la estructura de soporte mecánica. También pueden encontrarse los ficheros fuente en el repositorio donde se almacena el proyecto [29].

6.2. Fabricación de elementos mecánicos

En esta sección se introducirán los diferentes elementos mecánicos que se han fabricado además de la metodología seguida para ello.

6.2.1. Características mecánicas de las piezas

En la sección 6.1.3 se describen las características técnicas que tienen los diseños de las piezas en la fase de diseño. Sin embargo, también es preciso definir las características físicas que estas han de tener para poder ser fabricadas. Algunas características son las siguientes:

- Material de fabricación: para la fabricación de las piezas se usarán plásticos como ABS¹ o PLA². En el caso de que se trate de elementos rígidos se usará

¹El acrilonitrilo butadieno estireno, o ABS por sus siglas en inglés (*Acrylonitrile Butadiene Styrene*), es un plástico muy resistente al impacto muy utilizado en automoción y otros usos tanto industriales como domésticos.

²El ácido poliláctico o políácido láctico (PLA) es un polímero constituido por moléculas de ácido láctico, con propiedades semejantes a las del tereftalato de polietileno (PET) que se utiliza

aluminio.

- Sistemas de fijación: Para la fijación de elementos se usarán tornillos y tuercas.

6.2.2. Metodología de fabricación de piezas

Para la fabricación de las piezas principalmente se han usado dos técnicas, el **corte láser** y la **impresión 3D**. En las siguientes secciones se explican ambos métodos.

Impresión 3D de piezas

Para la impresión de las piezas 3D diseñadas, es necesario que los ficheros estén en un formato que el *software Slic3r* sea capaz de interpretar, típicamente estos formatos son **.obj** o **.stl**. Una vez las piezas están en este formato, se pueden introducir en *Slic3r* para generar los G-Code que se introducirán en la impresora para realizar los trabajos de impresión. En la configuración de *Slic3r* se deben tener en cuenta el tipo de filamento que se usará, la temperatura de extrusión y los parámetros de movimiento de la máquina, lo cuál determinará en gran medida la calidad del resultado final de la pieza impresa. Por otro lado, también se pueden configurar los soportes para ciertas zonas de las piezas como agujeros y voladizos.

Una vez configurado el G-Code, se puede introducir dicho fichero en la impresora 3D. Los métodos principales para enviar las instrucciones del G-Code a la impresora son la conexión serial con un PC o el uso de una tarjeta SD. Independientemente del método de carga del G-Code elegido, una vez puesta en marcha la impresión, se aplicará laca sobre la cama de cristal (en el caso de que la impresora no disponga de cama caliente) para que las primeras capas de la impresión se peguen correctamente a la superficie.

Finalmente, cuando la impresión llegue a su punto final, solo restará retirar de la impresora las piezas impresas con ayuda de una espátula o cuchilla. En el caso de que las piezas tengan puentes o soportes, puede ser necesaria la utilización de un alicate o de limas para corregir imperfecciones.

Corte láser de piezas

Para realizar los trabajos de corte láser, se coloca la pieza a cortar dentro de la máquina fijándola con ayuda de unos imanes y a continuación se realiza el ajuste de la potencia que aplicará el láser. Adicionalmente, se debe ajustar la distancia

para hacer envases, pero que además es biodegradable. Se degrada fácilmente en agua y óxido de carbono.

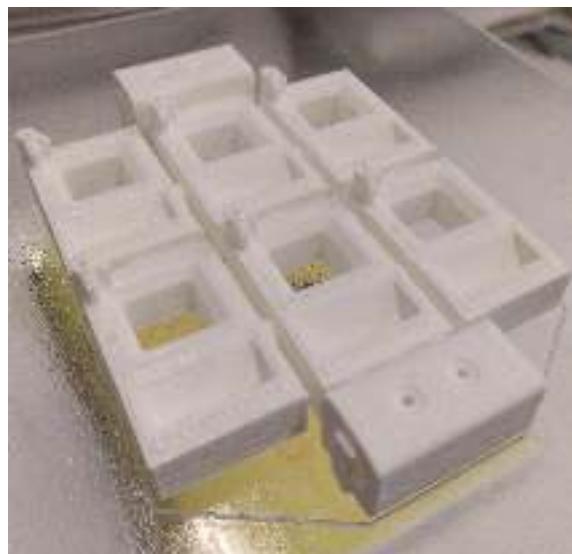


Figura 6.5: Módulo de calibración de cámaras impreso

entre el láser y el objeto que se quiere cortar, para evitar la dispersión del haz. Es importante cerrar la puerta de la máquina para evitar el escape de gases y el daño que puede producir el láser con el contacto visual. Aplicados todos estos ajustes, se pasa a ejecutar los trabajos de corte ajustando la velocidad de movimiento de la máquina si así se desea. Una vez terminadas las operaciones por parte de la cortadora, se podrá retirar la pieza y el sobrante de material.

6.2.3. Fabricación del calibrador de cámaras

Para la fabricación del calibrador de cámaras se usará una impresora 3D como la de la imagen de la Figura 2.4. El material utilizado para la fabricación es el PLA de color blanco extruido a una temperatura de 210°C. Los parámetros de configuración utilizados en *Slic3r* se han modificado para establecer una velocidad de movimiento de ejes baja para asegurar la precisión en la impresión de la pieza. El resultado de la impresión del calibrador de cámaras puede verse en la imagen de la Figura 6.5.

6.2.4. Fabricación de la estructura mecánica de soporte

La estructura mecánica de soporte se subdivide en dos partes fundamentales, la estructura básica y la mesa de soporte de la NJTX2. En las siguientes secciones se indican los detalles de la fabricación de cada una de las partes.



Figura 6.6: Piezas de estructura de soporte impresas

Fabricación de estructura básica de soporte

Para la fabricación de la estructura de soporte del sistema, se utilizan piezas impresas en 3D con PLA y por otro lado tubos de aluminio huecos (para que resulte ligera a la par que resistente). Los parámetros de configuración utilizados en *Slic3r* se han modificado para establecer una velocidad de movimiento de ejes moderada para asegurar que la precisión en la impresión de las piezas es buena a la par que su tiempo de impresión es moderado. En la imagen de la Figura 6.6 pueden verse algunas piezas de la estructura impresas.

Para el ensamblado de las diferentes piezas impresas se utilizan tubos de aluminio cortados según las medidas diseñadas. Para el corte de los tubos se usa un *corta tubos* para asegurar la precisión en el corte. Por otro lado, para introducir los tubos en las piezas se aplica presión hasta que se consigue insertar en esta. Para la fijación de las piezas móviles y algunos tubos, se usan tornillos de diámetro 2 mm. En la imagen de la Figura 6.7 puede verse la estructura básica una vez ensamblada.

Fabricación de mesa de soporte de NJTX2

Para la fabricación de la mesa de soporte de NJTX2 se usa el corte láser. El material utilizado para la fabricación es metacrilato transparente de color verde de 3 mm de grosor. Se han ajustado los parámetros de la máquina para una veloci-

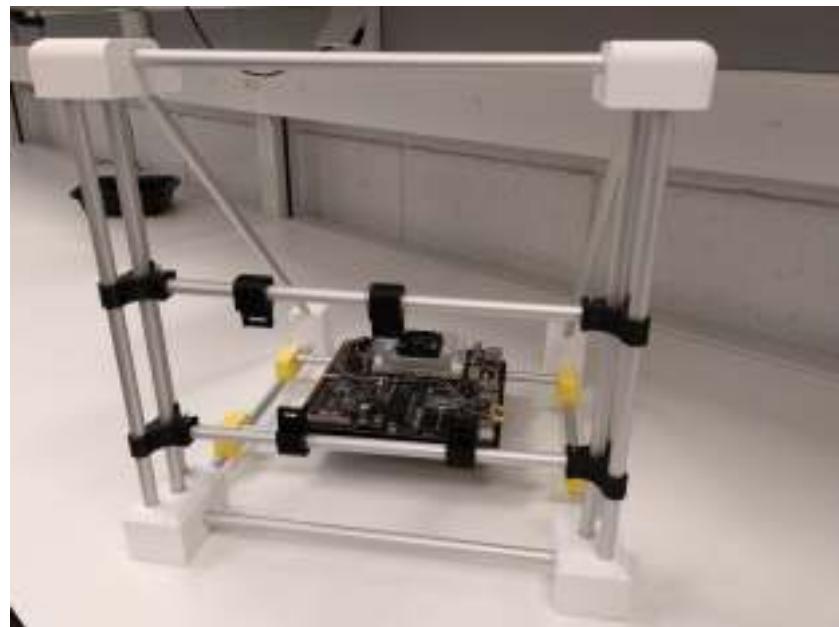


Figura 6.7: Estructura básica de soporte ensamblada



Figura 6.8: Mesa de soporte de NJTX2 fabricada

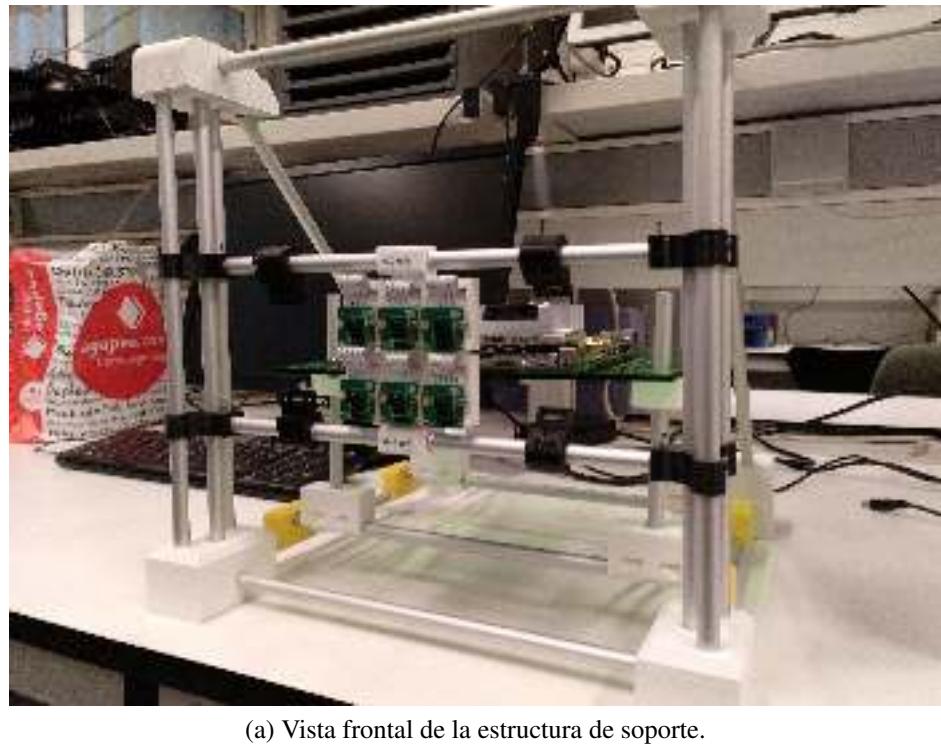
dad de avance baja para poder realizar los textos con la suficiente precisión. Las potencias aplicadas para las diferentes operaciones son:

- **Corte:** La potencia aplicada para el corte es 8W.
- **Grabado:** La potencia aplicada para el corte es 4W.

En la imagen de la Figura 6.8 puede verse el resultado de las operaciones de corte láser para la fabricación de la mesa de soporte de NJTX2. Una vez finalizado el proceso de fabricación de la mesa de soporte, es posible ensamblar la estructura completa.

6.3. Integración final

Una vez fabricados todos los elementos únicamente resta integrarlos entre sí para finalizar el montaje de la estructura al completo. En las imágenes de la Figura 6.9, puede verse la integración final de toda la estructura.



(a) Vista frontal de la estructura de soporte.



(b) Vista del calibrador de cámaras.

Figura 6.9: Estructura de soporte del sistema fabricada.

Capítulo 7

Configuración del Kernel y el *Device tree*

En este capítulo se explican todos los pasos seguidos para la correcta generación de los ficheros necesarios para la detección del módulo de cámara, con sensor IMX219, por parte de la Nvidia Jetson TX2. Todos los ficheros desarrollados en este capítulo podrán encontrarse en el repositorio donde se almacena el proyecto [29]. Para este propósito en primer lugar, se definen los pasos a seguir en el desarrollo para la implementación de los módulos de cámara.

7.1. Preparación de elementos

Antes de comenzar con el desarrollo es necesario contar con una serie de elementos básicos tanto de *software* como de *hardware* para la compilación del kernel y el *device tree* que permita la integración del módulo de cámara en el sistema. Se comienza con los elementos de *hardware* necesarios y se continuará enumerando los elementos de *software*.

7.1.1. *Hardware* necesario

Lo primero que se necesita es el kit de desarrollo de Nvidia Jetson TX2 como el mostrado en la imagen de la Figura 2.7. Por otro lado, se necesita un dispositivo que permita conectar las cámaras a la Jetson TX2, en el caso de este proyecto como se comenta al final del Capítulo 4, será el conector J20 de Auvidea [17]. Por último, se necesita los módulos de cámara que se implementarán en la placa. En el caso del conector J20 de Auvidea existen dos módulos compatibles, el módulo de cámara basado en el sensor OV5647 y el módulo de cámara basado en el sensor

IMX219. Para la implementación se opta por el módulo de cámara con el sensor IMX219 ya que presenta mejores prestaciones que el OV5647.

7.1.2. *Software necesario*

Con respecto al software necesario para la implementación de esta herramienta, será necesario lo siguiente:

- **Ubuntu 16.04 LTS**, para la programación y la carga del nuevo *kernel* en la Nvidia Jetson (tanto para la TX1 como para la TX2) es necesario contar con este sistema operativo. Esto es así debido a que el SDK desarrollado por Nvidia para la carga del código está específicamente desarrollado para este sistema operativo [11].
- **SDK Jetson TX1/TX2**, el Nvidia Jetpack proporciona las herramientas necesarias para la descarga y la modificación del código fuente de las Nvidia Jetson [18]. Por otro lado, también instalará en el sistema herramientas como TensorFlow o ejemplos para la renderización con CUDA que posteriormente podrían ser implementados en el kit de desarrollo.
- **Linaro ToolChain**, esta herramienta servirá para compilar el kernel de la Nvidia Jetson en una máquina externa y posteriormente cargarlo en la placa [24].

7.2. Desarrollo de software para la detección del sensor de cámara

Los pasos que típicamente se deben seguir para la correcta compilación del *kernel* e impletación del *device tree* y drivers para módulos de cámara, son los siguientes:

1. Preparar los ficheros fuente del *Device Tree*.
2. Modificar los archivos del *Device Tree* correctamente.
3. Modificar los ficheros fuente del módulo de cámara que se desee utilizar.
4. Configurar el Kernel para detectar los módulos que han sido añadidos.
5. Compilar el kernel.
6. Cargar el kernel en la Nvidia Jetson TX2 por medio de las herramientas proporcionadas en el Jetpack.

7. Cargar los **Device Tree Binary (DTB)** en la Jetson TX2 por medio de las herramientas proporcionadas en el Jetpack.
8. Testear el correcto funcionamiento del sistema.

Una vez se han definido los pasos a seguir, se pueden introducir algunos de ellos con un mayor nivel de detalle.

7.2.1. Preparación de ficheros fuente del *Device tree*

En general, en los sistemas Linux, existen dos formas de modificar el *kernel*. Los dos métodos son los siguientes:

- Mediante el **Archivo de configuración del kernel**: esto le indicará al *kernel* qué características y controladores deberían compilarse. También puede especificar qué controladores deben empaquetarse en el *kernel* y cuáles deben compilarse como módulos separados del *kernel* que podrán ser cargados con el *kernel* en funcionamiento mediante el comando “modprobe”. En el caso de este proyecto, se modificará esto para indicar al *kernel* que cree la compatibilidad con el controlador del sensor de cámara IMX219 y así poder cargarlo al arranque como un módulo del sistema [12].
- Mediante el **Device Tree Source (DTS)**: así es como puede configurar el *kernel* en el arranque para que se comporte de manera diferente en una placa frente a otra. Un claro ejemplo se puede encontrar con la Raspberry Pi y la Raspberry Pi Zero. Estas dos placas son muy diferentes a simple vista pero usan el mismo módulo del *kernel* gracias a que implementan diferentes ficheros para el *device tree*.

Como se ha comentado, se busca agregar soporte para el módulo de cámara con sensor IMX219 (Cámara Raspberry Pi Modelo 2.1). Debido a que no existe un mecanismo para reconocer las cámaras CSI mientras el *kernel* se está ejecutando, se debe modificar el **DTS** para indicar al *kernel* cómo reconocer los módulos. Esto es muy diferente con respecto al funcionamiento de un USB, por ejemplo, porque en el caso del USB existe un mecanismo estándar que el *kernel* puede usar para reconocer cuando se inserta un nuevo dispositivo, para el CSI esto no es así. Por lo tanto, cada placa está configurada con un archivo interno conocido como **DTB (Device Tree Binary)**. Este **DTB** es una versión compilada de los archivos **DTS (Device Tree Source)** y **DTSi (Device Tree Source Include)**.

En el fichero de configuración DTS utilizado para reconocer el módulo de cámara IMX219, se deben describir los siguientes aspectos:

- A qué puertos CSI están conectadas las cámaras.

- Cómo se deben configurar las cámaras usando i^2c .
- Qué modos de visualización tienen las cámaras (Resoluciones soportadas por el sensor).
- Cómo configurar la arquitectura de entrada de video del TX2 (Cómo se espera que se codifique el vídeo).

Todo esto debe recogerse en un fichero DTS. Existe la posibilidad de comenzar este fichero desde cero, pero para la gran mayoría de casos puede usarse el archivo DTS predeterminado utilizado para los kits de desarrollo de Nvidia. En el caso de que se quiera obtener un mayor control sobre este procedimiento véase [35]. Para facilitar el proceso, se optará por crear el fichero DTS partiendo de los códigos creados por Nvidia. Para la obtención de los ficheros fuente véase [19].

7.2.2. Modificación de ficheros fuente del *device tree*

Una vez se han descargado los ficheros fuente del *kernel* de la NJTX2, se modificará el fichero base de los DTS para incluir los ficheros que se desarrollen. Principalmente se deben desarrollar dos ficheros, uno que configurará la(s) cámara(s) al nivel de *hardware* y el otro para configurar la arquitectura interna de la Interfaz de video de la NJTX2. Dentro del fichero de configuración es necesario configurar una serie de parámetros para lograr la detección de la(s) cámara(s). Los módulos son los siguientes:

- En la interfaz con el *kernel* de la NJTX2:
 - Componente NVCSI: Este componente servirá para configurar las líneas CSI para comunicarse con las cámaras correctamente.
 - Componente VI: Configura la salida del NVCSI de forma directa a las líneas del ISP o directamente sobre la memoria del dispositivo.
 - Componente I2C: Configuración de la salida de la cámara a vídeo.
- GPIOs:
 - Configurar los GPIOs para que posean cierto comportamiento como por ejemplo, resetear la cámara, encenderla o apagar sus reguladores.
 - Resetear las líneas para poder así resetear las cámaras.
 - Incorporación de las direcciones i^2c de la multiplexación.
- Configuración de dispositivos *hardware* en la NJTX2:

7.2. DESARROLLO DE SOFTWARE PARA LA DETECCIÓN DEL SENSOR DE CÁMARA113

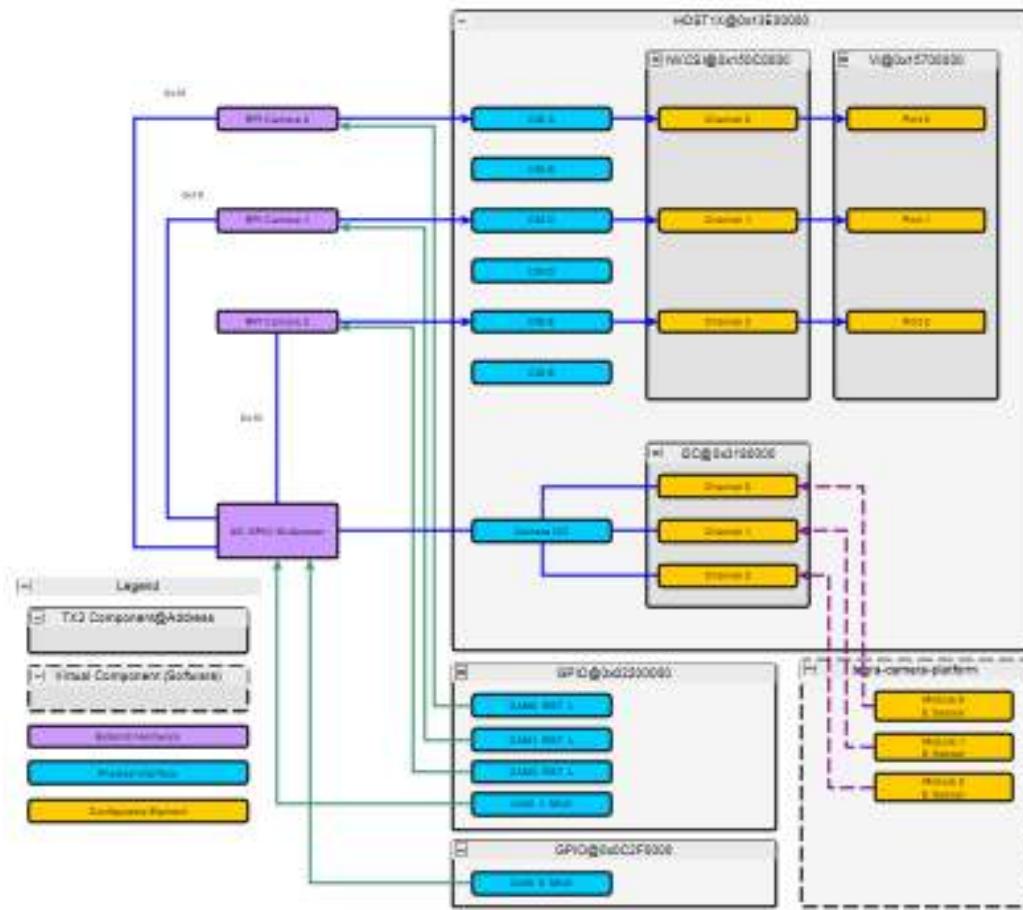


Figura 7.1: Diagrama de integración de los módulos que componen el DTS [28]

- Describir la localización física de la cámara.
- Configurar el ISP para procesar las imágenes.
- Describir otros elementos que puedan hacer falta a la cámara, como por ejemplo el enfoque o el control del obturador.

Para comprender mejor todo lo anterior, se ha ilustrado la idea principal en el diagrama de la Figura 7.1.

Todas las cámaras de Raspberry Pi tienen la misma dirección i^2c , por lo que será necesario disponer de una forma de distinguir una cámara de otra. La manera más fácil es con un multiplexor o un traductor de direcciones, como el que se encuentra implementado en el módulo puente J20.

En alto nivel si se observa el diagrama de la Figura 7.1 donde se ejemplifica el uso de 3 cámaras de Raspberry Pi en la NJTX2, los datos de las cámaras



Figura 7.2: Montaje para test de líneas con el módulo de depuración

de Raspberry Pi llegan a través de los canales CSI representados en azul. Solo es necesario un bus CSI de dos canales en este caso así que se toma el CSI A, si necesitaran cuatro canales, se reservarían dos bloques o canales CSI. A continuación, en el componente **NVCSI** se extraerá la imagen sin formato del protocolo CSI y enviará los datos al componente **VI**. Este componente denominado **Video Input** es esencialmente un enrutador de video que enruta el video al ISP o directamente a la memoria del dispositivo. Si la cámara de Raspberry Pi tiene un procesador de señal de imagen interno para convertir la imagen a una imagen RGB útil, se indicará en la configuración del **VI** que enrute los datos de imagen a la memoria de forma directa. En el caso del módulo de cámara basado en el sensor IMX219, este no tiene un ISP, por lo que se usará el ISP que posee la Jetson TX2. El proceso de enrutar los datos de video después del componente **VI** se realiza dentro del entorno de usuario cuando el kernel se inicia, por lo que la configuración finaliza una vez que el **VI** está conectado correctamente, de esta forma el usuario ya tiene acceso a los datos. Comúnmente el acceso a los datos se realiza mediante los dispositivos `/dev/videox`.

Para la realización de pruebas en la salida de las líneas del conector J22 de la Nvidia Jetson TX2, puede utilizarse el módulo de depuración fabricado, visto en la sección 5. Puede verse un test con el analizador lógico en la imagen de la Figura 7.2.

Configuración de direcciones i^2c

Para la configuración de los ficheros DTS, es necesario configurar unas direcciones específicas para los módulos de cámara. Estas direcciones cambian en función del módulo de cámara utilizado o de la programación de la EEPROM de dichos módulos. Por otro lado, se encuentran las direcciones de los buses de la Nvidia Jetson TX2 que son las mostradas en la Tabla 7.1.

| Bus | Alias | Dirección i^2c del bus |
|------|---------------------|--------------------------|
| I2C0 | gen1 i2c | i2c@3160000 |
| I2C1 | gen2 i2c | i2c@c240000 |
| I2C2 | cam i2c | i2c@3180000 |
| I2C3 | dp aux channel1 i2c | i2c@3190000 |
| I2C4 | pwr i2c | bpmp i2c |
| I2C5 | dp aux channel0 i2c | i2c@31b0000 |
| I2C6 | gen7 i2c | i2c@31c0000 |
| I2C7 | gen8 i2c | i2c@c250000 |
| I2C8 | gen9 i2c | i2c@31e0000 |

Tabla 7.1: Direcciones i^2c y alias en la Nvidia Jetson TX2

En el caso de usar el módulo puente J20 para la interfaz entre las cámaras y la Nvidia Jetson TX2, las direcciones i^2c serán traducidas para poder conectar 6 cámaras usando solamente 3 buses. Esencialmente se traduce la dirección de una de las cámaras para cada bus, de esta forma se consigue la conexión de dos cámaras iguales por bus teniendo que usar la mitad de buses i^2c . La traducción para los posibles módulos de cámara que pueden conectarse al módulo puente J20 son:

- Cámara de Raspberry Versión 1.3 (Sensor OV5647):
 - Dirección i^2c normal: 0x36
 - Dirección traducida: 0x34
- Cámara de Rapsberry Versión 2.1 (Sensor IMX219):
 - Dirección i^2c normal: 0x10
 - Dirección traducida: 0x12

Teniendo en cuenta las direcciones anteriores, y que solamente se usarán 3 buses i^2c de la NJTX2 para la implementación de las cámaras, la configuración de direcciones en el DTS es la mostrada en la Tabla 7.2.

| Bus | Alias de cámara | Dirección i^2c del dispositivo |
|-------------|------------------------|--|
| i2c@c240000 | imx219 a | 0x10 |
| | imx219 b | 0x12 |
| i2c@3180000 | imx219 c | 0x10 |
| | imx219 d | 0x12 |
| i2c@c250000 | imx219 e | 0x10 |
| | imx219 f | 0x12 |

Tabla 7.2: Configuración de direcciones i2c en los ficheros DTS para la conexión de 6 cámaras

Una vez conocidas las direcciones i^2c , puede verse un ejemplo de configuración de un módulo de cámara en el *device tree* en el siguiente fragmento de código extraído del *device tree*:

```

1 i2c@c240000 {
2     imx219_a@10 {
3         compatible = "nvidia,imx219";
4         /* I2C device address */
5         reg = <0x10>;
6         status = "okay";
7         /* V4L2 device node location */
8         devnode = "video0";
9
10        clocks = <&tegra_car TEGRA186_CLK_EXTPERIPH1>,
11        <&tegra_car TEGRA186_CLK_PLLP_OUT0>;
12        clock-names = "extperiph1", "pll_p_grtba";
13
14        /* Define any required hw resources needed by driver */
15        /* ie. clocks, io pins, power sources */
16
17        mclk = "extperiph1";
18        clock-frequency = <24000000>;
19        reset-gpios = <&tegra_main_gpio CAM0_RST_L
20        GPIO_ACTIVE_HIGH>;
21        pwdn-gpios = <&tegra_main_gpio CAM0_PWDN GPIO_ACTIVE_HIGH
22        >;
23        vana-supply = <&en_vdd_cam_hv_2v8>;
24        vif-supply = <&en_vdd_cam>;
25
26        refclk_hz = <27000000>;
27        /* Physical dimensions of sensor */
28        physical_w = "5.095";
29        physical_h = "4.930";
30        /* Sensor Model */
31        sensor_model = "imx219";
32        avdd-reg = "vana";
33        dvdd-reg = "vdig";
34        iovdd-reg = "dovdd";

```

```

33     post_crop_frame_drop = "0";
34     mode0 {
35         mclk_khz = "24000";
36         num_lanes = "2";
37         tegra_sinterface = "serial_a";
38         discontinuous_clk = "yes";
39         dpcm_enable = "false";
40         cil_settletime = "0";
41         active_w = "3280";
42         active_h = "2464";
43         pixel_t = "bayer_rggb";
44         readout_orientation = "0";
45         line_length = "3446";
46         inherent_gain = "1";
47         mclk_multiplier = "25";
48         pix_clk_hz = "182400000";
49         min_gain_val = "1.0";
50         max_gain_val = "16.0";
51         min_hdr_ratio = "1";
52         max_hdr_ratio = "64";
53         min_framerate = "1.462526";
54         max_framerate = "21";
55         min_exp_time = "13";
56         max_exp_time = "683709";
57         embedded_metadata_height = "2";
58     };

```

En el fragmento de código mostrado, se configura la dirección del bus i^2c dentro de la NJTX2 y su línea CSI. En el caso de este fragmento, se trata de la cámara que se conecta con la línea CSI-A. Además, se indica la dirección i^2c del módulo de cámara, 0x10 en este caso. Por último, se establece toda la configuración para el funcionamiento del módulo de cámara, las velocidades de reloj, las direcciones de los puertos GPIO, el formato de los datos de entrada/salida,etc.

Todos los ficheros desarrollados para la configuración del *device tree* pueden encontrarse en el repositorio donde se almacena el proyecto [29] como parches de *software*, para ser aplicado directamente sobre el código fuente de la NJTX2.

7.2.3. Modificación de drivers para módulos de cámara y compilación del kernel

Para la correcta detección de los módulos de cámara, es necesario disponer de los drivers necesarios que gestionen el correcto funcionamiento de los sensores. Para ello, Nvidia proporciona en su código fuente diferentes versiones de drivers para los sensores más comunes. Una vez se han obtenido los drivers para los

módulos de cámara que se pretendan implementar, en el caso de este proyecto el IMX219, se podrán incluir en los ficheros fuente para ser compilados.

Una vez modificados los DTS y añadidos los drivers de los módulos de cámara, se puede compilar el kernel de la Nvidia Jetson TX2 y posteriormente puede cargarse la imagen .iso generada por medio del puerto serie en la placa. Para más información sobre este proceso véase [19].

7.3. Detección de dispositivos usando el conector J20

Una vez realizada la carga de los ficheros binarios de configuración del *device tree* y los drivers del módulo de cámara, el sistema operativo será capaz de reconocer los dispositivos de cámara que se conecten en las líneas CSI especificadas en la Tabla 7.2. Para la detección de los dispositivos es posible usar el comando *i2cdetect*, aunque previamente hay que activar los GPIO del módulo puente J20. En ocasiones los GPIO del módulo puente J20 tienen problemas para detectar los flancos de bajada. Por ello, es necesario reiniciarlos de forma manual (poniendo todos los bits a cero) para que se puedan seguir usando las cámaras con normalidad. Para más información sobre la activación y el reinicio de los GPIO del módulo puente J20 puede consultar la hoja de datos del Anexo C.7 [17].

En la Tabla 7.3 pueden verse las direcciones i^2c de los dispositivos de cámara además de la correspondencia con los conectores presentes en el módulo puente Auvidea J20.

| Conejor J20 | Cámara | Bus i^2c | Dirección i^2c de la cámara | Línea CSI |
|-------------|--------|------------|-------------------------------|-----------|
| J1-1 | OV5647 | 1 | 0x36 | CSI-A |
| | IMX219 | 1 | 0x10 | CSI-A |
| J1-2 | OV5647 | 2 | 0x36 | CSI-C |
| | IMX219 | 2 | 0x10 | CSI-C |
| J1-3 | OV5647 | 7 | 0x36 | CSI-E |
| | IMX219 | 7 | 0x10 | CSI-E |
| J2-1 | OV5647 | 1 | 0x36 | CSI-B |
| | IMX219 | 1 | 0x10 | CSI-B |
| J2-2 | OV5647 | 2 | 0x36 | CSI-D |
| | IMX219 | 2 | 0x10 | CSI-D |
| J2-3 | OV5647 | 7 | 0x36 | CSI-F |
| | IMX219 | 7 | 0x10 | CSI-F |

Tabla 7.3: Detección de dispositivos i^2c en el módulo puente J20 de Auvidea [17]

Una vez detectados los dispositivos correctamente, las cámaras podrán ser utilizadas por parte del sistema operativo.

Capítulo 8

Adquisición y procesado de imágenes

En este capítulo se explican todos los pasos seguidos para la adquisición de imágenes y su procesado para la generación de mapas de disparidad. Una vez definidos los elementos de interfaz necesarios para el reconocimiento de las cámara en el Capítulo 7, en este capítulo se tratan todos los aspectos relacionados con el uso de las mismas desde el sistema operativo. Para ello, el primer paso es la adquisición de las imágenes.

8.1. Adquisición de imágenes

Para la adquisición de las imágenes se han utilizado principalmente dos utilidades de *software*, *Gstreamer-1.0* y *v4l2*. En las siguientes secciones se explica el funcionamiento básico de cada una de ellas y sus características principales.

8.1.1. *Gstreamer-1.0*

GStreamer es una biblioteca para construir gráficos de componentes de manejo de medios. Las aplicaciones que admite abarcan desde la reproducción simple de Ogg/Vorbis, la transmisión de audio/video hasta el procesamiento complejo de audio y de video [16]. Por ello, se ha utilizado *Gstreamer-1.0* para la adquisición de imágenes en tiempo real desde las cámaras. En la imagen de la Figura 8.1 se muestra un ejemplo. Además se ha utilizado la posibilidad de mezcla entre vídeos e imágenes que proporciona *Gstreamer-1.0* para la calibración física de las cámaras.

Puede encontrar todos los *scripts* desarrollados para las capturas con *Gstreamer-1.0* en el repositorio donde se almacena el proyecto [29].



Figura 8.1: Streaming de vídeo en tiempo real de 6 cámaras con *Gstreamer-1.0*



Figura 8.2: Captura realizada con *v4l2* desde una de las cámaras del array (Imagen Posprocesada)

8.1.2. V4L2

Video4Linux(V4L2) es una colección de controladores de dispositivos y una API para admitir la captura de video en tiempo real en sistemas Linux. Es compatible con muchas cámaras web USB, sintonizadores de TV y dispositivos relacionados, estandarizando su salida, por lo que se puede agregar soporte de video para aplicaciones [33]. En este proyecto se ha utilizado *V4L2* para la adquisición de las imágenes por parte de cada uno de los módulos de cámara. Este *software* permite el uso de las diferentes resoluciones para realizar las capturas de las imágenes (todas ellas programadas en los módulos de driver vistos en el Capítulo 7) En la imagen de la Figura 8.2 puede verse un ejemplo de una captura realizada usando *V4L2*.

Para la integración de los algoritmos de estimación de disparidad y otras utilidades, se usarán las capturas realizadas con *v4l2*. Puede encontrar los *scripts*

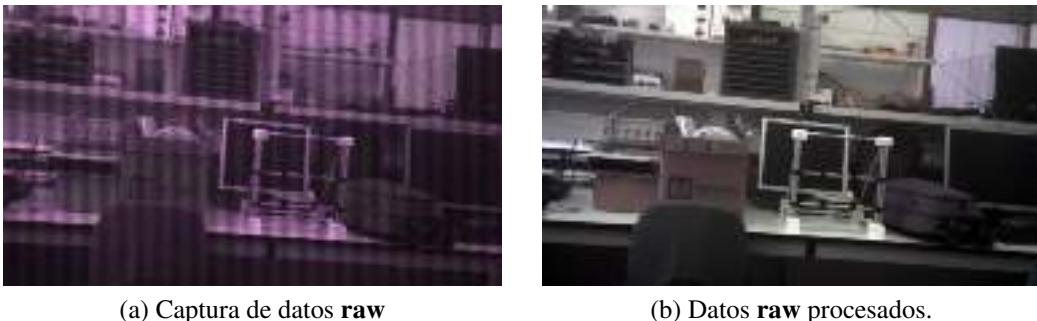


Figura 8.3: Comparativa entre datos **raw** y datos tratados.

desarrollados para las capturas usando *V4l2* en el repositorio donde se almacena el proyecto [29].

8.2. Tratamiento de los datos *raw*

Para la utilización de las imágenes capturadas usando *V4l2* es necesario realizar un posprocesado de la imagen. Como puede observarse en las imágenes de las Figura 8.3 la imagen original capturada por *v4l2* (Figura 8.3a) presenta colores diferentes a los de la escena real y además tiene una gran cantidad de ruido. Para la corrección de esta imagen, se ha programado un script de *Python* en el que se usan varias utilidades de tratamiento de imágenes de *OpenCV* [27]. En primer lugar, se aplica un método para equilibrar los canales de color en base a los valores máximos y mínimos de cada uno de ellos, de esta forma se consigue un mejor contraste en la imagen. Por otro lado, se realiza un filtrado de la imagen para eliminar el ruido presente en esta evitando eliminar detalles de la propia imagen (la ventana utilizada para el filtrado es pequeña). El resultado del posprocesado de la imagen *raw* puede verse en la imagen de la Figura 8.3b.

Pueden encontrarse los *scripts* para el tratamiento de los datos *raw* en el repositorio donde se almacena el proyecto [29].

8.3. Calibración de cámaras

Para que la aplicación de algoritmos para el cálculo de mapas de disparidad sea posible, es importante que las cámaras que estén colocadas en el *array* tengan un punto en el cuál coincidan las direcciones de apuntado. Por otro lado, también es necesaria la calibración por *software* de las cámaras para el cálculo de sus



Figura 8.4: Montaje para ajustar el ángulo de apuntado de las cámaras

parámetros intrínsecos¹ y extrínsecos. En las siguientes secciones se tratarán más en profundidad tanto la calibración física para ajustar el apuntado de las cámaras como la calibración por *software* para el cálculo de los parámetros intrínsecos y extrínsecos.

8.3.1. Calibración física de las cámaras

Para el ajuste del ángulo de apuntado de las cámaras presentes en el *array*, se ha colocado un panel con un patrón de tablero de ajedrez frente a las cámaras. Se puede ver el montaje en la imagen de la Figura 8.4.

Para la modificación del ángulo de las cámaras se usa el calibrador de cámaras fabricado en el Capítulo 6. Adicionalmente, se ha diseñado un sistema de *streaming* de vídeo con una rejilla sobrepuerta sobre el vídeo en tiempo real. Con ayuda de la rejilla se realiza el ajuste sobre uno de los puntos del tablero de ajedrez en la imagen, normalmente se ajusta el apuntado sobre una de las esquinas de un cuadrado. Modificando el ajuste de los tornillos de la estructura de calibración el ángulo de las cámaras varía, y esto es apreciable en el *stream* de vídeo ya que se verá que los objetos se mueven en la imagen. En las imágenes de la Figura 8.5 se pueden ver tanto el ajuste de la rejilla como la modificación de ajuste de los tornillos.

¹Los parámetros intrínsecos son aquellos que describen el funcionamiento de una cámara, son por ejemplo, la distancia focal, el punto principal y el centro óptico.



(a) Ajuste de rejilla.



(b) Modificación de ángulo de apuntado de cámara mediante los tornillos de calibración.

Figura 8.5: Ajuste de apuntado de cámaras.

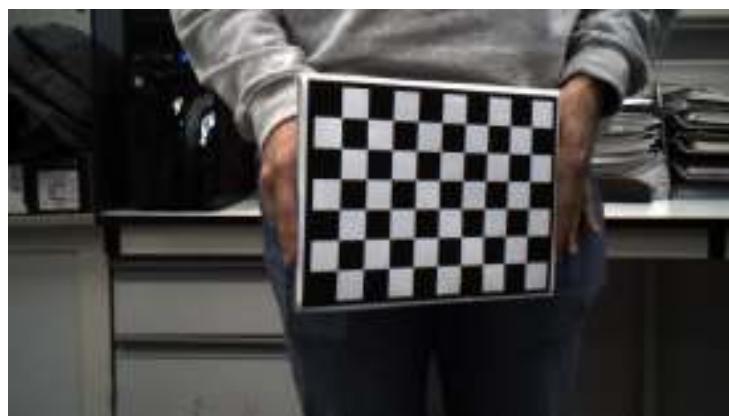


Figura 8.6: Patrón para el cálculo de parámetros de cámaras

Es importante durante este proceso asegurar que las cámaras apuntan exactamente al mismo punto del tablero. Cuanto mejor sea la calibración realizada, mejor será el resultado obtenido a la hora de aplicar los algoritmos.

8.3.2. Calibración por software

La calibración de las cámaras mediante *software* se basa en encontrar los parámetros internos, externos y posibles distorsiones que presentan las cámaras. Esencialmente los parámetros más relevantes a calcular en la calibración por *software* se recogen en la siguiente expresión:

$$\begin{pmatrix} f_x & 0 & C_x \\ 0 & f_y & C_y \\ 0 & 0 & 1 \end{pmatrix} \quad (8.1)$$



(a) Imágenes de calibración, Vista 5 , CAM 0. (b) Imágenes de calibración, Vista 11 , CAM 0.

Figura 8.7: Ejemplo de capturas de calibración.

Por otro lado los coeficientes de distorsión de la cámara se recogen de la siguiente forma:

$$[K_1, K_2, p_1, p_2, K_3] \quad (8.2)$$

Todos estos parámetros se pueden obtener mediante imágenes obtenidas por medio de las cámaras y un proceso de ajuste de mínimos con un modelo matemático de la cámara. Para el cálculo de los parámetros se han elaborado *scripts* de *Python* en los que se usan las herramientas de calibración de *openCV* para este fin. Antes de ejecutar el *script* de calibración es necesario capturar las imágenes de un patrón de calibración (imagen de la Figura 8.6) para que ser leídas por el *script* y así calcular los parámetros. El patrón de calibración utilizado en este caso, es un patrón de tablero de ajedrez de 10 x 7 cuadros de 28 milímetros cada uno. Para la calibración, no se tienen en cuenta los cuadros de los extremos por lo que realmente el patrón a detectar será de 9 x 6 cuadros.

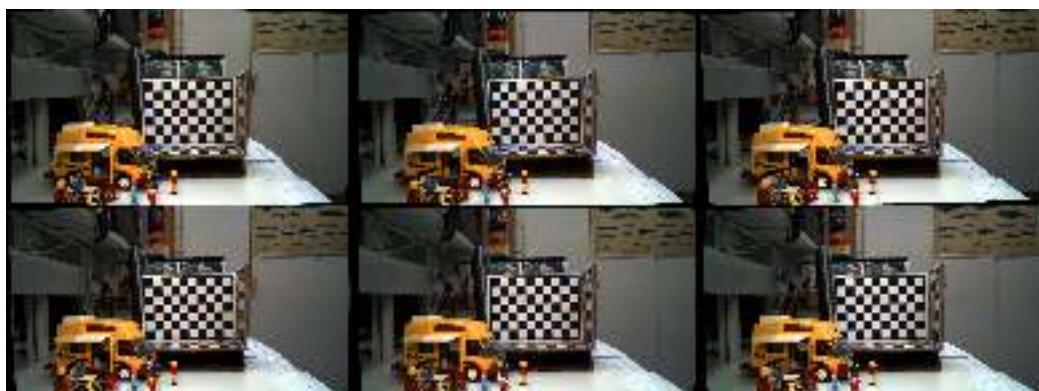
Después se toman varias capturas del patrón desde distintas posiciones (moviendo el patrón y no las cámaras). De esta manera con varias capturas del patrón por cada una de las cámaras, el algoritmo será capaz de calcular los parámetros intrínsecos y extrínsecos de cada cámara. En las imágenes de la Figura 8.7 pueden verse algunos ejemplos de captura de imágenes de calibración.

Durante la ejecución del *script* para el cálculo de los parámetros intrínsecos y extrínsecos de las cámaras se usa lo que se conoce como la detección de *corners* o esquinas de los cuadros del tablero de calibración [26]. En la imagen de la Figura 8.8 puede verse un ejemplo de detección de corners. Gracias a su detección el algoritmo implementado en las librerías de calibración de *openCV* es capaz de calcular los parámetros de cámara que se han mencionado.

Pueden encontrarse todos los *scripts* desarrollados para la calibración de las cámaras en el repositorio donde se almacena el proyecto [29].



Figura 8.8: Detección de corners

Figura 8.9: *Light Field* generado

8.4. Generación del *Light Field*

El siguiente paso antes de la estimación del mapa de disparidad es la generación de lo se conoce como *Light Field*. Un *Light Field* se compone de las diferentes imágenes obtenidas por las cámaras una vez eliminada la distorsión. Además, en las imágenes se realiza la homografía con una imagen de referencia dentro del *array*, en este proyecto se usará la imagen central inferior como imagen de referencia para las homografías. De esta manera, el *Light Field* estaría preparado para su utilización en la estimación de mapas de disparidad.

Para la generación del *Light Field* se ha desarrollado un *script* que puede encontrarse en el repositorio donde se almacena el proyecto [29].

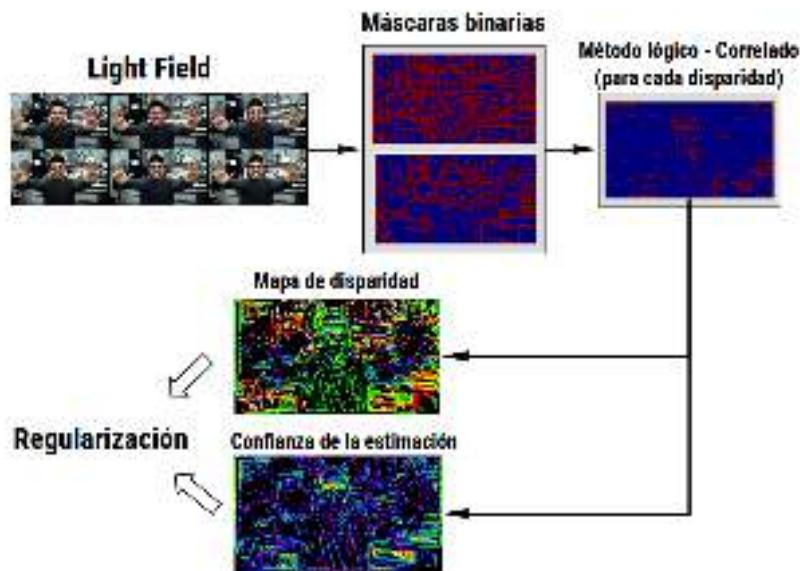


Figura 8.10: Proceso de estimación del mapa de distancias

8.5. Estimación del mapa de disparidad

El último paso del tratamiento de las imágenes es la estimación de lo que se conoce como mapa de disparidad. Este mapa no es más que una representación gráfica de los objetos que se encuentran más cerca o más alejados de las cámaras usando una escala de color para ello.

En el diagrama de la Figura 8.10 puede observarse el proceso seguido para la generación de los mapas de disparidad. Este proceso se inicia con un *Light Field* que previamente se ha generado cogiendo las capturas realizadas por las cámaras, modificándolas para eliminar las distorsiones y realizando la homografía con respecto a la cámara de referencia. Posteriormente se generan las máscaras binarias de cada una de las vistas. Para cada vista se generan dos máscaras binarias, una de ellas conserva más estructura de la imagen que la otra. Seguidamente, se aplica un algoritmo lógico-correlado en el que esencialmente se estiman los objetos que se encuentran dentro de cierta disparidad. En el caso del diagrama de la Figura 8.10 la máscara que se muestra para el algoritmo lógico-correlado es la resultante para disparidad cero, lo cual indica qué objetos se estiman que se encuentran en el plano de calibración de las cámaras. Es lo que se denomina plano de disparidad cero. Este proceso se repite para cada una de las disparidades que contengan las imágenes. Finalmente, integrando todas las máscaras de disparidad generadas y sus hipótesis se obtiene una estimación del mapa de disparidad como se puede

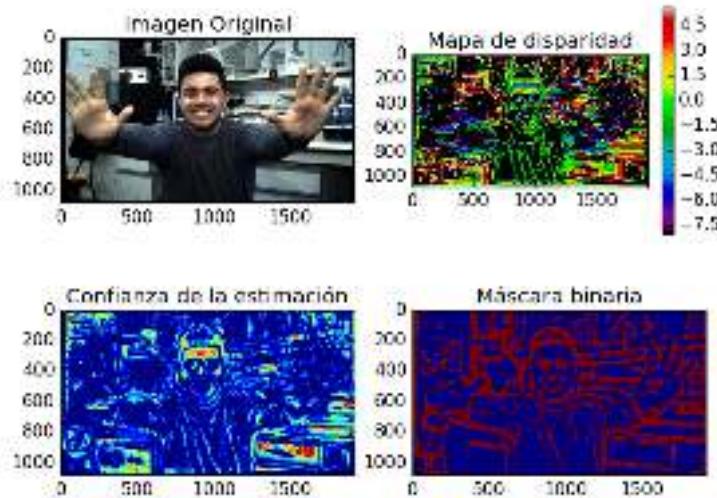


Figura 8.11: Resultados de la estimación de distancias

ver en la Figura 8.11. Por otro lado, se puede estimar la posible veracidad de los datos generados en la estimación del mapa de disparidad. Esto es lo que se ha denominado como confianza. Las zonas más coloridas del mapa de confianza son aquellas en las que hay mayor certeza de haber realizado una buena estimación de la disparidad. En la Figura 8.11, se puede observar el resultado obtenido para la estimación de la disparidad además de las máscaras generadas para el cálculo. Como se puede observar en la imagen, existen algunas zonas de color negro. Esto es así debido a que el algoritmo no ha podido realizar una estimación o por el contrario la estimación realizada tiene confianza cero, por lo tanto ese punto se toma como no calculado y representa en negro para no confundirlo con los puntos que si han sido calculados.

Capítulo 9

Resultados de la estimación de disparidad

En este breve capítulo se muestran los resultados obtenidos en las pruebas realizadas con el *array* de cámaras para la estimación de mapas de disparidad. Para la correcta comprobación de los resultados, se han construido unas escenas de las cuales se conocen las distancias desde las cámaras hasta los objetos que son visibles en las imágenes capturadas. De esta forma, se podrá interpretar mucho mejor si la información extraída del mapa de disparidad es correcta o por contra es errónea.

Las escenas se han iluminado y montado como se muestra en la imagen de la Figura 9.1. Es importante disponer de una buena iluminación para las escenas debido a que si esta es oscura, las imágenes presentarán una gran cantidad de ruido. Para la medida de la posición de los objetos dentro de la escena se ha graduado la mesa donde se han colocado para tener una noción de la distancia entre las cámaras y los objetos.

Principalmente se han elegido dos escenas para el test de rendimiento de la estimación de disparidad del *array* de cámaras. Es importante mencionar que para el proceso de estimación del mapa de disparidad no se ha aplicado ningún proceso de regularización del mismo por lo que el resultado obtenido en cada una de las escenas, aún tiene margen de mejora. Los resultados se muestran en las siguientes dos secciones.

9.1. Resultados para la primera escena

La primera escena para la estimación del mapa de disparidad está compuesta por juguetes de pequeño tamaño para comprobar cuál es la resolución en la estimación de la distancia a la que se encuentran los objetos pequeños. En las



Figura 9.1: Montaje e iluminación de una escena

imágenes de la Figura 9.2 se puede observar el resultado de la estimación (Figura 9.2b) además de una vista superior desde donde se ve la posición relativa entre los objetos presentes en la escena(Figura 9.2a).

Como puede observarse en los resultados de la Figura 9.2b las distancias estimadas coinciden con la posición relativa de los objetos dentro de la escena vistos en la imagen de las Figura 9.2a. Las zonas más uniformes dentro de la imagen como la mesa, o las puertas del armario del fondo no es posible calcularlas ya que carecen de textura. Por esta razón, tanto las puertas del armario como la mesa aparecen en color negro (no calculados) ya que el método utilizado para la estimación de las disparidades no es capaz de determinar una hipótesis. Por otro lado, la confianza de la estimación es bastante alta en todas las zonas, lo cuál implica que la estimación realizada es buena.

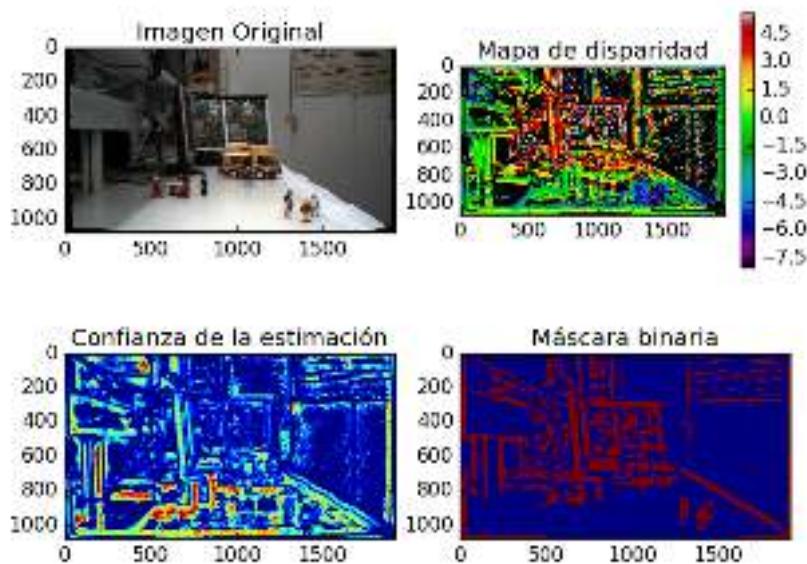
9.2. Resultados para la segunda escena

La segunda escena para la estimación del mapa de disparidad está compuesta por juguetes de gran tamaño en comparación con los de la primera escena. De esta forma, con una mayor resolución de la estructura de los objetos en la imagen, se quiere comprobar cuál es rendimiento de la estimación de la disparidad. En las imágenes de la Figura 9.3 se puede observar el resultado de la estimación realizada para la disparidad (Figura 9.2b) además de una vista superior desde donde se ve la posición relativa entre los objetos presentes en la escena (Figura 9.2a).

Como puede observarse en los resultados de la Figura 9.3b, la estimación es muy similar a la realizada para la escena 1. La resolución de esta imagen es ligeramente mejor que para el primer caso debido a que las máscaras realizadas



(a) Visualización de la escena 1.

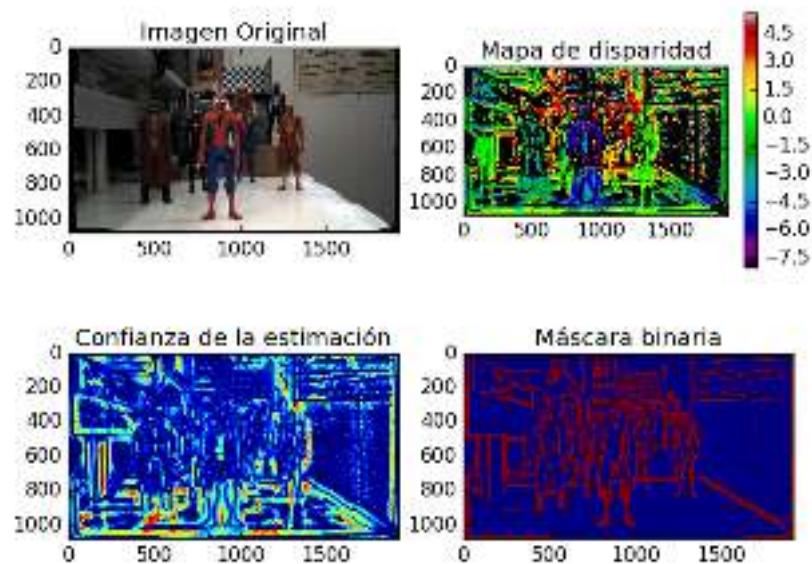


(b) Estimación de disparidad de la escena 1.

Figura 9.2: Resultados de la estimación de la escena 1.



(a) Visualización de la escena 2.



(b) Estimación de disparidad de la escena 2.

Figura 9.3: Resultados de la estimación de la escena 2.

alcanzan un mayor nivel de detalle en objetos grandes que en objetos de poco tamaño. Se puede comprobar que las distancias estimadas coinciden con la posición relativa de los objetos dentro de la escena vistos en la imagen de las Figura 9.3a. Al igual que ocurrió para la primera escena, las zonas más uniformes dentro de la imagen como la mesa, o las puertas del armario del fondo no es posible calcularlas ya que carecen de textura y se estiman como puntos de confianza cero o no calculados.

Capítulo 10

Conclusiones

Como conclusión general, en la realización de este proyecto se han cumplido los siguientes objetivos:

1. Se ha diseñado una placa electrónica capaz de adquirir los datos procedentes de un sensor CMOS.
2. Se ha diseñado una interfaz electrónica para la conexión de múltiples cámaras a la **Nvidia Jetson TX2**.
3. Se ha fabricado prototipos de placas de circuito impreso que permiten una mejor comprensión de la plataforma utilizada.
4. Se ha desarrollado la integración hardware-software para la **Nvidia Jetson TX2**.
5. Se ha desarrollado el software de adquisición de imágenes con el uso de *v4l2-src* y *Gstreamer-1.0* .
6. Se ha diseñado una estructura mecánica capaz de sustentar el sistema de adquisición.
7. Se ha diseñado un sistema para calibrar el ángulo de apuntado de las cámaras.
8. Se ha fabricado por medio de técnicas de prototipado rápido un prototipo funcional del diseño mecánico realizado.
9. Se han integrado los diferentes elementos diseñados.
10. Se ha desarrollado la calibración de la array de cámaras por medio de software.

11. Se ha comprobado el funcionamiento de la array de cámaras desarrollada con un algoritmo de estimación simple.

Como se puede deducir de la relación anterior, se han cumplido todos los objetivos propuestos al inicio del proyecto. El desarrollo de la adaptación multicámaras para la Nvidia Jetson TX2 dota de gran potencial a esta plataforma debido a su estructura de procesamiento. En este proyecto se han desarrollado todos los elementos necesarios para dicha adaptación, preparando así la plataforma para su uso de captura y procesamiento de imágenes. La principal ventaja de este desarrollo es lo asequible que resulta su montaje a nivel económico, lo cual resulta interesante cuando se habla de *arrays* de cámaras de bajo coste. Por esta razón, el desarrollo de este proyecto se presenta como una aportación competitiva a nivel económico en este campo. Por otro lado, gracias al uso de tecnologías como la impresión 3D se ha conseguido reducir el coste de la fabricación de elementos como el calibrador de cámaras.

Como conclusión general, se ha podido comprobar que el rendimiento de la estimación de disparidad utilizando el sistema diseñado es lo suficientemente bueno como para integrar este sistema en proyectos donde se incorporen cámaras 3D de bajo coste.

Líneas abiertas

Durante el desarrollo de este proyecto se han encontrado una serie de posibles líneas abiertas sobre las que continuar su desarrollo. La principal de todas ellas es la posibilidad de procesar los datos procedentes de las cámaras con el uso de la GPU incorporada en la Nvidia Jetson TX2. Con la adecuada programación de módulos de *Gstreamer-1.0* se podrían procesar en tiempo real las imágenes procedentes de las cámaras, mostrando así la estimación de disparidad en tiempo real. Esta posibilidad presenta una serie de retos que podrían afrontarse en futuros trabajos de fin de grado de la titulación de Ingeniería informática, ya que presenta una alta carga de trabajo en el desarrollo de *software*.

Conclusions

As a general conclusion, the following objectives have been achieved in the implementation of this project:

1. An electronic board has been designed capable of acquiring data from a CMOS sensor.
2. An electronic interface has been designed for connecting multiple cameras to the **Nvidia Jetson TX2**.
3. Prototypes of printed circuit boards have been produced to allow a better understanding of the platform used.
4. The hardware-software integration for the **Nvidia Jetson TX2** was developed.
5. The image acquisition software has been developed with the use of *v4l2-src* and *Gstreamer-1.0* .
6. A mechanical structure has been designed to support the acquisition system.
7. A system has been designed to calibrate the pointing angle of the cameras.
8. A functional prototype of the mechanical design has been produced using rapid prototyping techniques.
9. The various elements designed have been integrated.
10. Calibration of the camera array has been developed using software.
11. The camera array developed has been tested with a simple estimation algorithm .

As can be seen from the above list, all the objectives proposed at the beginning of the project have been met. The development of the multi-camera adaptation for the Nvidia Jetson TX2 gives great potential to this platform due to its processing

structure. In this project all the necessary elements for the adaptation have been developed, thus preparing the platform for its use in image capture and processing. The main advantage of this development is the affordability of its economical installation, which is interesting when talking about low-cost camera arrays. For this reason, the development of this project is presented as an economically competitive contribution in this field. On the other hand, thanks to the use of technologies such as 3D printing, the following has been achieved reduce the cost of manufacturing elements such as the camera calibrator.

As a general conclusion, it has been found that the performance of the disparity estimation using the designed system is good enough to integrate this system in projects where low-cost 3D cameras are incorporated.

Open lines

During the development of this project a series of possible open lines have been found on which to continue its development. The main one of all of them is the ability to process data from cameras using the GPU built into the Nvidia Jetson TX2. With the right module programming of *Gstreamer-1.0*, images from the cameras could be processed in real time, showing the real-time disparity estimate. This possibility presents a number of challenges that could be addressed in future end-of-degree work on the degree in Computer Engineering, as it presents a high burden of work in software development.

Bibliografía

- [1] Varios autores. *camotics*. URL: <http://camotics.org>.
- [2] Varios autores. *Debian*. URL: <https://www.debian.org/>.
- [3] Varios autores. *FlatCAM*. URL: <http://flatcam.org/>.
- [4] Varios autores. *FreeCAD*. URL: <http://freecadweb.org/>.
- [5] Varios autores. *FreeRouting*. URL: <http://www.freerouting.net/>.
- [6] Varios autores. *Inkscape*. URL: <https://inkscape.org/es/>.
- [7] Varios autores. *Kile - an Integrated LaTeX Environment*. URL: <http://kile.sourceforge.net/>.
- [8] Varios autores. *LaTeX Reference*. URL: <http://maths.cnam.fr/Membres/wilk/MathMax/help/LaTeX/latex.html>.
- [9] Varios autores. *python*. URL: <https://www.python.org/>.
- [10] Varios autores. *slic3r*. URL: <http://slic3r.org/about>.
- [11] Varios autores. *Ubuntu*. URL: <https://www.ubuntu.com/>.
- [12] *Building External Modules - Linux Kernel*. URL: <https://www.kernel.org/doc/Documentation/kbuild/modules.txt> (visitado 26-06-2018).
- [13] *CAFADIS - University of La Laguna*. URL: <http://www.cafadis.ull.es/> (visitado 16-06-2018).
- [14] Nvidia Corporation. *NVIDIA Jetson Camera Module Hardware Design Guide*. Nov. de 2017. URL: <https://developer.nvidia.com/>.
- [15] Paul E. Debevec y Jitendra Malik. «Recovering High Dynamic Range Radiance Maps from Photographs». En: *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH '97. New York, NY, USA: ACM Press/Addison-Wesley Publishing Co., 1997, págs. 369-378. ISBN: 9780897918961. DOI: [10.1145/258734.258884](https://doi.org/10.1145/258734.258884). URL: <https://doi.org/10.1145/258734.258884> (visitado 17-06-2018).

- [16] *GStreamer: open source multimedia framework.* URL: <https://gstreamer.freedesktop.org/> (visitado 26-06-2018).
- [17] *J20 – Auvidea.* en-US. URL: <https://auvidea.com/j20/> (visitado 25-06-2018).
- [18] *JetPack.* en. Oct. de 2014. URL: <https://developer.nvidia.com/embedded/jetpack> (visitado 25-06-2018).
- [19] *Jetson TX1 Source Code | SDK for Jetson TX1 | RidgeRun Developer.* en. URL: http://developer.ridgerun.com/wiki/index.php?title=Compiling_Tegra_X1/X2_source_code#Build_Kernel (visitado 26-06-2018).
- [20] *Kits de desarrollo y Módulos Jetson para sistemas embebidos.* es-es. URL: <https://www.nvidia.es/autonomous-machines/embedded-systems-dev-kits-modules/> (visitado 16-06-2018).
- [21] Walter Lain. *Kicad: Electronic CAD libraries.* URL: <http://smisioto.no-ip.org/elettronica/kicad/kicad-en.htm>.
- [22] Leslie Lamport. *LaTeX – A document preparation system.* URL: <https://latex-project.org/>.
- [23] Marc Levoy y Pat Hanrahan. «Light Field Rendering». En: *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH '96. New York, NY, USA: ACM, 1996, págs. 31-42. ISBN: 9780897917469. DOI: [10.1145/237170.237199](https://doi.acm.org/10.1145/237170.237199). URL: <http://doi.acm.org/10.1145/237170.237199> (visitado 19-06-2018).
- [24] *Linaro Toolchain.* en. URL: <https://www.linaro.org/about/> (visitado 25-06-2018).
- [25] *MIPI Camera Serial Interface 2 (MIPI CSI-2).* URL: <https://www.mipi.org/specifications/csi-2> (visitado 16-06-2018).
- [26] *OpenCV: Harris Corner Detection.* URL: https://docs.opencv.org/3.4/dc/d0d/tutorial_py_features_harris.html (visitado 26-06-2018).
- [27] *OpenCV library.* URL: <https://opencv.org/> (visitado 26-06-2018).
- [28] *Raspberry Pi Camera on TX2.* URL: http://cospandesign.github.io/linux_tx2_kernel_driver/2017/12/15/tx2-rpi-camera-port.html (visitado 26-06-2018).
- [29] Iván Rodríguez. *Adap_multicamara_NJTX2: Desarrollo de la adaptación para multicámara de la Nvidia Jetson TX2.* Trabajo de fin de Máster de Iván Rodríguez Méndez dirigido por Fernando Luis Rosa González. original-date: 2018-06-30T11:53:57Z. Jun. de 2018. URL: https://github.com/tidus747/Adap_multicamara_NJTX2 (visitado 30-06-2018).

- [30] Richard Szeliski. *Computer Vision: Algorithms and Applications*. en. Texts in Computer Science. London: Springer-Verlag, 2011. ISBN: 9781848829343. URL: [//www.springer.com/us/book/9781848829343](http://www.springer.com/us/book/9781848829343) (visitado 17-06-2018).
- [31] The KiCad Team. *KiCad EDA: A Cross Platform and Suite, Open Source Electronics Design Automation*. URL: <http://docs.kicad-pcb.org/es/kicad.html>.
- [32] Linus Torvalds y community. *Git: a Free and Open Source Distributed Version Control System*. URL: <https://git-scm.com/>.
- [33] Video4Linux. en. Page Version ID: 814990339. Dic. de 2017. URL: <https://en.wikipedia.org/w/index.php?title=Video4Linux&oldid=814990339> (visitado 26-06-2018).
- [34] Bennett Wilburn y col. «High Performance Imaging Using Large Camera Arrays». En: *ACM SIGGRAPH 2005 Papers*. SIGGRAPH '05. New York, NY, USA: ACM, 2005, págs. 765-776. DOI: [10.1145/1186822.1073259](https://doi.acm.org/10.1145/1186822.1073259). URL: <http://doi.acm.org/10.1145/1186822.1073259> (visitado 17-06-2018).
- [35] Writing a Kernel Driver. URL: http://cospandesign.github.io/linux_kernel_driver/2016/04/15/kernel-driver.html (visitado 26-06-2018).

**ESCUELA DE DOCTORADO
Y ESTUDIOS DE POSGRADO**

Trabajo de Fin de Máster

**Desarrollo de la adaptación para multicámara
de la Nvidia Jetson TX2**

TOMO II

Pliego de condiciones y presupuesto

Titulación: Máster en Ingeniería Industrial

Estudiante: Iván Rodríguez Méndez

Tutor: Fernando Luis Rosa González

25 de junio de 2018

Capítulo 11

Pliego de condiciones

La principal condición para la realización de este proyecto consiste en que el desarrollo de la adaptación multicámaras debe usar *software* y *hardware* libre, y además debe ser funcional al final de su desarrollo. Otras condiciones para el desarrollo del proyecto son las siguientes:

- Emplear un sistema operativo Linux.
- Utilizar *KiCad EDA* para el desarrollo del diseño electrónica.
- Usar *Kile* para la redacción y documentación del proyecto.
- Emplear Git para la compartición de archivos y documentos del proyecto.
- Emplear FreeCAD para el diseño de piezas mecánicas.
- Utilizar Python para el desarrollo de software.

Por otro lado, todo el hardware desarrollado durante el presente proyecto será libre.

Capítulo 12

Presupuesto del proyecto

El presupuesto del proyecto asciende a un total de 6976,87 €. De los cuales 6000 € son gastos de personal y 976,87 € son gastos de material y de fabricación. En la tabla 12.1, se descompone dicho presupuesto, donde se indica los elementos implicados, su cantidad, su precio unitario (si procede) y el precio total.

| Elemento | Cantidad (ud) | Precio (€/ud) | Total (€) |
|---|---------------|---------------|-----------|
| Componentes electrónicos | 2 | - | 12,97 |
| Placa de circuito impreso | 1 | 5,50 | 5,50 |
| Gastos iniciales de fabricación electrónica | - | - | 18,00 |
| Gastos iniciales de fabricación mecánica | - | - | 10,00 |
| Bobinas para impresora 3D | 2 | - | 35,00 |
| Componentes mecánicos | 2 | - | 15,75 |
| Nvidia Jetson TX2 | 1 | 569,00 | 569,00 |
| Auvidea J20 | 1 | 158,25 | 158,25 |
| Cámaras Rpi V2.1 | 6 | 25,40 | 152,40 |
| Precio por hora de diseño electrónico | 100 | 20,00 | 2000,00 |
| Precio por hora de diseño mecánico | 100 | 20,00 | 2000,00 |
| Precio por hora de desarrollo de software | 100 | 20,00 | 2000,00 |

Tabla 12.1: Descomposición del presupuesto del proyecto

El precio de la placa de circuito impreso, de los gastos iniciales de fabricación, el montaje de componentes son los cargos realizados estimados para la fabricación y montaje de la placas. Por otro lado, para la fabricación de la mecánica únicamente se ha tenido en cuenta el coste de material y el coste inicial de la fabricación. Por último, se ha establecido un precio bruto de 20 €/h por el trabajo de diseño

y desarrollo de las diferentes partes del proyecto, que incluye el coste de su labor así como los gastos de Seguridad Social e impuestos.

El desglose del coste total de los componentes mecánicos utilizados se puede ver en la tabla 12.2.

| Grupo | Componente | Distribuidor | Unidades | €/ud | € |
|-----------|--------------------------|--------------|----------|--------|-------|
| Tornillos | Tornillos de 2 mm | Leroy Merlin | 40 | 0,1125 | 4,50 |
| Tubos | Tubos de aluminio de 1 m | Leroy Merlin | 5 | 2,25 | 11,25 |

Tabla 12.2: Precio detallado de los componentes mecánicos

El desglose del coste total de los componentes electrónicos utilizados se puede ver en la tabla 12.3, donde se puede ver la cantidad de cada uno de ellos que se necesita para el diseño y el precio total en función del coste unitario establecido por el distribuidor correspondiente.

| Grupo | Componente | Distribuidor | Unidades | €/ud | € |
|---|------------------|--------------|----------|-------|-------|
| Conecotor | QTH-060-01-L-D-A | RS | 1 | 10,50 | 10,50 |
| Conecotor HARWIN Placa a placa Recto | M20-9990345 | Farnell | 29 | 0,085 | 2,47 |

Tabla 12.3: Precio detallado de los componentes electrónicos

**ESCUELA DE DOCTORADO
Y ESTUDIOS DE POSGRADO**

Trabajo de Fin de Máster

**Desarrollo de la adaptación para multicámara
de la Nvidia Jetson TX2**

**TOMO III
Anexos**

Titulación: Máster en Ingeniería Industrial

Estudiante: Iván Rodríguez Méndez

Tutor: Fernando Luis Rosa González

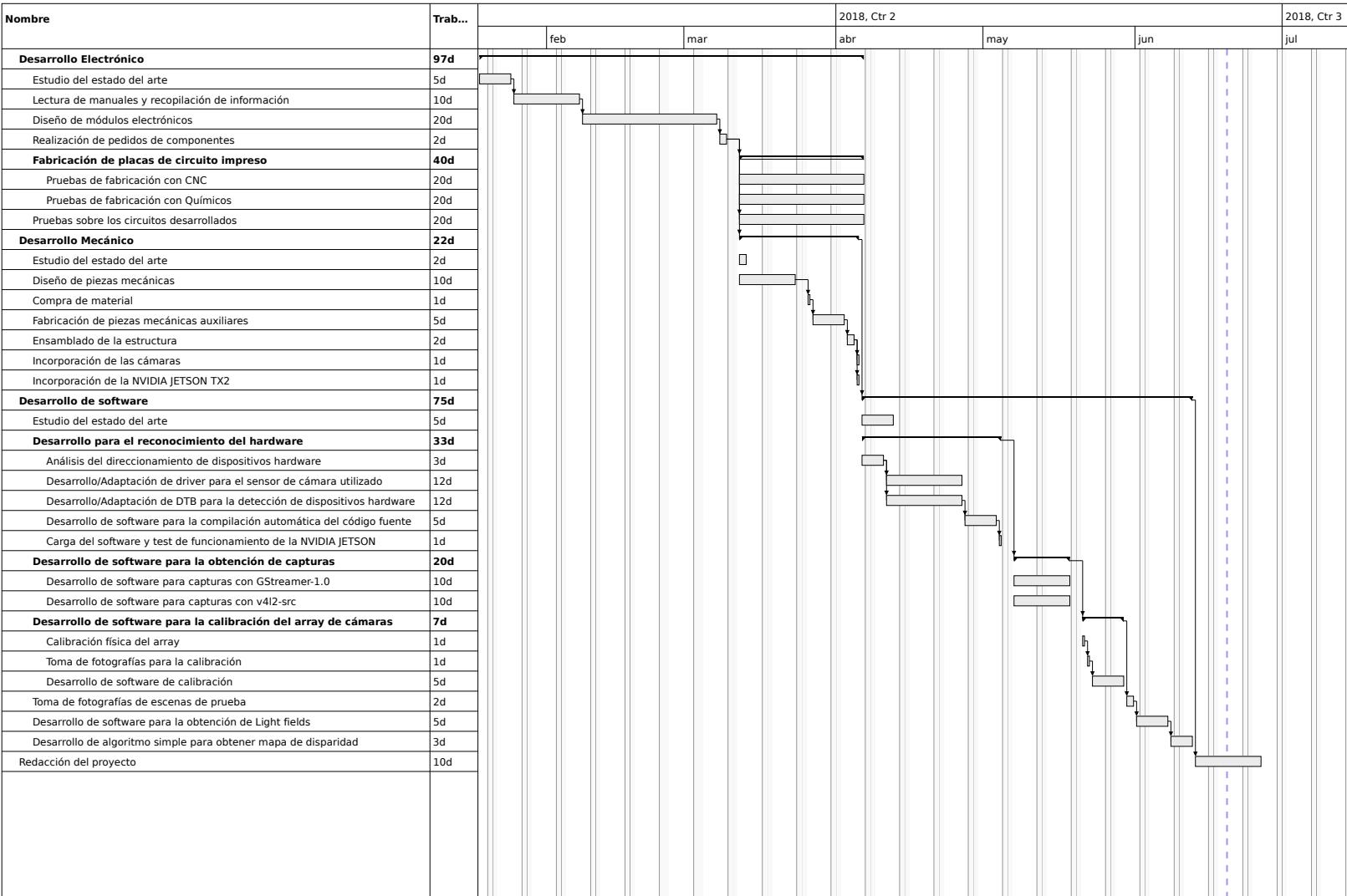
25 de junio de 2018

Anexo A

Cronología del proyecto

En este anexo se presenta la cronología que ha seguido el proyecto en su desarrollo. Se mostrará un diagrama de Gantt indicando las precedencias entre tareas y paquetes de trabajo, y una tabla donde se indican las fechas de inicio y fin de cada una de ellas.

A.1. Diagrama de Gantt



A.2. CUADRO DE TEMPORALIZACIÓN

153

A.2. Cuadro de temporalización

| WBS | Nombre | Inicio | Fin | Trabajo | Duración | Desperdicio | Coste | Asignado a | % Completado |
|-------|---|---------------|---------------|------------|------------|-------------|----------|------------|--------------|
| 1 | Desarrollo Electrónico | ene 18 | abr 6 | 97d | 57d | 57d | 0 | | 0 |
| 1.1 | Estudio del estado del arte | ene 18 | ene 24 | 5d | 5d | | 0 | | 0 |
| 1.2 | Lectura de manuales y recopilación de información | ene 25 | feb 7 | 10d | 10d | | 0 | | 0 |
| 1.3 | Diseño de módulos electrónicos | feb 8 | mar 7 | 20d | 20d | | 0 | | 0 |
| 1.4 | Realización de pedidos de componentes | mar 8 | mar 9 | 2d | 2d | | 0 | | 0 |
| 1.5 | Fabricación de placas de circuito impreso | mar 12 | abr 6 | 40d | 20d | 37d | 0 | | 0 |
| 1.5.1 | Pruebas de fabricación con CNC | mar 12 | abr 6 | 20d | 20d | 37d | 0 | | 0 |
| 1.5.2 | Pruebas de fabricación con Químicos | mar 12 | abr 6 | 20d | 20d | 37d | 0 | | 0 |
| 1.6 | Pruebas sobre los circuitos desarrollados | mar 12 | abr 6 | 20d | 20d | 57d | 0 | | 0 |
| 2 | Desarrollo Mecánico | mar 12 | abr 5 | 22d | 19d | | 0 | | 0 |
| 2.1 | Estudio del estado del arte | mar 12 | mar 13 | 2d | 2d | 17d | 0 | | 0 |
| 2.2 | Diseño de piezas mecánicas | mar 12 | mar 23 | 10d | 10d | | 0 | | 0 |
| 2.3 | Compra de material | mar 26 | mar 26 | 1d | 1d | | 0 | | 0 |
| 2.4 | Fabricación de piezas mecánicas auxiliares | mar 27 | abr 2 | 5d | 5d | | 0 | | 0 |
| 2.5 | Ensamblado de la estructura | abr 3 | abr 4 | 2d | 2d | | 0 | | 0 |
| 2.6 | Incorporación de las cámaras | abr 5 | abr 5 | 1d | 1d | | 0 | | 0 |
| 2.7 | Incorporación de la NVIDIA JETSON TX2 | abr 5 | abr 5 | 1d | 1d | | 0 | | 0 |
| 3 | Desarrollo de software | abr 6 | jun 12 | 75d | 48d | | 0 | | 0 |
| 3.1 | Estudio del estado del arte | abr 6 | abr 12 | 5d | 5d | 43d | 0 | | 0 |
| 3.2 | Desarrollo para el reconocimiento del hardware | abr 6 | may 4 | 33d | 21d | | 0 | | 0 |
| 3.2.1 | Análisis del direccionamiento de dispositivos hardware | abr 6 | abr 10 | 3d | 3d | | 0 | | 0 |
| 3.2.2 | Desarrollo/Adaptación de driver para el sensor de cámara utilizado | abr 11 | abr 26 | 12d | 12d | 6d | 0 | | 0 |
| 3.2.3 | Desarrollo/Adaptación de DTB para la detección de dispositivos hardware | abr 11 | abr 26 | 12d | 12d | | 0 | | 0 |
| 3.2.4 | Desarrollo de software para la compilación automática del código fuente | abr 27 | may 3 | 5d | 5d | | 0 | | 0 |
| 3.2.5 | Carga del software y test de funcionamiento de la NVIDIA JETSON | may 4 | may 4 | 1d | 1d | | 0 | | 0 |
| 3.3 | Desarrollo de software para la obtención de capturas | may 7 | may 18 | 20d | 10d | | 0 | | 0 |
| 3.3.1 | Desarrollo de software para capturas con GStreamer-1.0 | may 7 | may 18 | 10d | 10d | | 0 | | 0 |
| 3.3.2 | Desarrollo de software para capturas con v4l2-src | may 7 | may 18 | 10d | 10d | | 0 | | 0 |
| 3.4 | Desarrollo de software para la calibración del array de cámaras | may 21 | may 29 | 7d | 7d | | 0 | | 0 |
| 3.4.1 | Calibración física del array | may 21 | may 21 | 1d | 1d | | 0 | | 0 |
| 3.4.2 | Toma de fotografías para la calibración | may 22 | may 22 | 1d | 1d | | 0 | | 0 |
| 3.4.3 | Desarrollo de software de calibración | may 23 | may 29 | 5d | 5d | | 0 | | 0 |
| 3.5 | Toma de fotografías de escenas de prueba | may 30 | may 31 | 2d | 2d | | 0 | | 0 |
| 3.6 | Desarrollo de software para la obtención de Light fields | jun 1 | jun 7 | 5d | 5d | | 0 | | 0 |
| 3.7 | Desarrollo de algoritmo simple para obtener mapa de disparidad | jun 8 | jun 12 | 3d | 3d | | 0 | | 0 |
| 4 | Redacción del proyecto | jun 13 | jun 26 | 10d | 10d | | 0 | | 0 |

Anexo B

Datasheets de los Integrados y componentes utilizados

Las siguientes hojas de características pertenecen a los distintos circuitos integrados y componentes que se han empleado en el circuito.

B.1. Conector Molex FPC 30 pin



This document was generated on 02/02/2015

PLEASE CHECK WWW.MOLEX.COM FOR LATEST PART INFORMATION

Part Number:

54104-3031

Status:

Active

Overview:

FFC-FPC (SMT)

Description:

0.50mm Pitch Easy-On™ Type FFC/FPC Connector, Right Angle, Surface Mount, ZIF, Top Contact Style, 30 Circuits, 0.1µm Gold (Au) Plating

Documents:

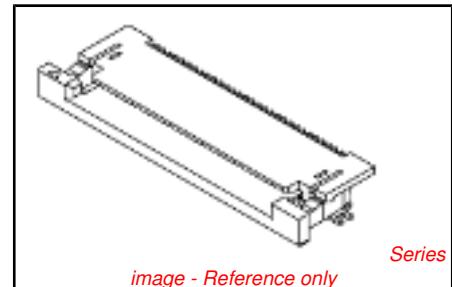
3D Model

Drawing (PDF)

Product Specification PS-54104-071 (PDF)

[Packaging Specification SPK-54104-001 \(PDF\)](#)

[RoHS Certificate of Compliance \(PDF\)](#)



Series

image - Reference only

Agency Certification

CSA

LR19980

UL

E29179

General

Product Family

FFC/FPC Connectors

Series

54104

Overview

[FFC-FPC \(SMT\)](#)

Product Name

Easy-On™

UPC

822348689088

Physical

Circuits (Loaded)

30

Color - Resin

Natural

Contact Position

Top

Durability (mating cycles max)

20

Entry Angle

90° Angle

Flammability

94V-0

Mated Height

2.00mm

Material - Metal

Phosphor Bronze

Material - Plating Mating

Gold

Material - Plating Termination

Gold

Material - Resin

High Temperature Thermoplastic

Net Weight

355.810/mg

Number of Rows

1

Orientation

Right Angle

PCB Locator

No

PCB Retention

Yes

Packaging Type

Embossed Tape on Reel

Pitch - Mating Interface

0.50mm

Plating min - Mating

0.102µm

Plating min - Termination

0.051µm

Polarized to PCB

Yes

Stackable

No

Surface Mount Compatible (SMC)

N/A

Temperature Range - Operating

-20°C to +85°C

Termination Interface: Style

Surface Mount

Wire/Cable Type

FFC/FPC

Electrical

Current - Maximum per Contact

0.5A

Voltage - Maximum

50V

Material Info

Reference - Drawing Numbers

EU RoHS

ELV and RoHS Compliant

REACH SVHC

Contains SVHC: No

Low-Halogen Status

Not Low-Halogen

China RoHS



Need more information on product environmental compliance?

Email productcompliance@molex.com

For a multiple part number RoHS Certificate of Compliance, [click here](#)

Please visit the Contact Us section for any non-product compliance questions.

Search Parts in this Series

54104Series

Mates With

Molex Premo-Flex FFC. Be sure to match like plating

B.2. Ceramic wire wound Inductors

Ceramic Wire Wound Inductors

AISC-0402
 RoHS / RoHS II Compliant

1.19 x 0.64 x 0.66mm

▷ FEATURES:
▷ APPLICATIONS:

- Exceptional high Q and SRF for high frequency applications
- Excellent DCR and current carrying characteristics
- Small size suitable for surface mounting
- Epoxy coating protects wound wire

- Widely applied in mobile phones such as GSM, CDMA, PDC, etc.
- Bluetooth, W-LAN, Broadband Network
- High Frequency Communication Circuits
- Video cameras, liquid crystal television, and other electronic devices

ELECTRICAL SPECIFICATIONS:

| Part Number AISC-0402- Inductance Code | L (nH) | Tolerance | Q Min | L/Q Test Freq (MHz) | SRF Min (MHz) | RDC Max (Ω) | IDC Max (mA) |
|---|--------|------------|----------|---------------------------|---------------------|-------------------|--------------------|
| 1N0 | 1 | J, K, S | 13 | 250 | 12700 | 0.045 | 1360 |
| 1N2 | 1.2 | J, K, S | 13 | 250 | 12000 | 0.060 | 1300 |
| 1N8 | 1.8 | J, K, S | 13 | 250 | 11500 | 0.070 | 1040 |
| 1N9 | 1.9 | J, K, S | 16 | 250 | 11300 | 0.070 | 1040 |
| 2N0 | 2 | J, K, S | 16 | 250 | 11100 | 0.070 | 1040 |
| 2N2 | 2.2 | J, K, S | 18 | 250 | 10800 | 0.070 | 960 |
| 2N4 | 2.4 | J, K, S | 18 | 250 | 10500 | 0.070 | 960 |
| 2N7 | 2.7 | J, K, S | 13 | 250 | 10400 | 0.120 | 640 |
| 3N0 | 3 | J, K, S | 20 | 250 | 7000 | 0.066 | 840 |
| 3N3 | 3.3 | G, J, K, S | 20 | 250 | 7000 | 0.066 | 840 |
| 3N6 | 3.6 | G, J, K, S | 20 | 250 | 6800 | 0.066 | 840 |
| 3N9 | 3.9 | G, J, K, S | 20 | 250 | 6000 | 0.066 | 840 |
| 4N3 | 4.3 | G, J, K, S | 20 | 250 | 6000 | 0.091 | 700 |
| 4N7 | 4.7 | G, J, K, S | 20 | 250 | 4775 | 0.083 | 800 |
| 5N1 | 5.1 | G, J, K, S | 23 | 250 | 5800 | 0.083 | 800 |
| 5N6 | 5.6 | G, J, K, S | 23 | 250 | 5800 | 0.083 | 760 |
| 6N2 | 6.2 | G, J, K, S | 23 | 250 | 5800 | 0.083 | 760 |
| 6N8 | 6.8 | G, J, K | 20 | 250 | 5800 | 0.083 | 680 |
| 7N3 | 7.3 | G, J, K | 25 | 250 | 6000 | 0.130 | 570 |
| 7N5 | 7.5 | G, J, K | 25 | 250 | 5800 | 0.100 | 680 |
| 8N2 | 8.2 | G, J, K | 25 | 250 | 4400 | 0.100 | 680 |
| 8N7 | 8.7 | G, J, K | 25 | 250 | 4200 | 0.100 | 680 |
| 9N0 | 9 | G, J, K | 25 | 250 | 4160 | 0.100 | 680 |
| 9N5 | 9.5 | G, J, K | 21 | 250 | 4000 | 0.162 | 600 |
| 10N | 10 | G, J, K | 21 | 250 | 3900 | 0.200 | 480 |
| 11N | 11 | G, J, K | 26 | 250 | 3680 | 0.120 | 640 |
| 12N | 12 | G, J, K | 26 | 250 | 3600 | 0.120 | 640 |
| 13N | 13 | G, J, K | 26 | 250 | 3450 | 0.185 | 440 |
| 15N | 15 | G, J, K | 26 | 250 | 3280 | 0.170 | 560 |
| 16N | 16 | G, J, K | 26 | 250 | 3100 | 0.220 | 560 |
| 18N | 18 | G, J, K | 26 | 250 | 3100 | 0.230 | 480 |
| 19N | 19 | G, J, K | 26 | 250 | 3040 | 0.200 | 480 |
| 20N | 20 | G, J, K | 26 | 250 | 3000 | 0.250 | 420 |
| 22N | 22 | G, J, K | 26 | 250 | 2800 | 0.250 | 400 |
| 23N | 23 | G, J, K | 26 | 250 | 2720 | 0.250 | 400 |
| 24N | 24 | G, J, K | 26 | 250 | 2700 | 0.300 | 400 |
| 27N | 27 | G, J, K | 26 | 250 | 2480 | 0.300 | 400 |
| 30N | 30 | G, J, K | 25 | 250 | 2350 | 0.300 | 400 |
| 33N | 33 | G, J, K | 25 | 250 | 2350 | 0.350 | 400 |

ABRACON IS
ISO9001:2008
CERTIFIED
 **ABRACON**
CORPORATION
Visit www.abracon.com for Terms & Conditions of Sale
30332 Esperanza, Rancho Santa Margarita, California 92688
tel 949-546-8000 | fax 949-546-8001 | www.abracon.com
Revised: 07.29.14

B.3. EEPROM AT24C01D

Atmel®

AT24C01D and AT24C02D

I²C-Compatible (2-wire) Serial EEPROM
1-Kbit (128 x 8) or 2-Kbit (256 x 8)

DATASHEET

Features

- Low Voltage Operation
 - 1.7V (V_{CC} = 1.7V to 3.6V)
- Internally Organized 128 x 8 (1K) or 256 x 8 (2K)
- I²C-Compatible (2-wire) Serial Interface
 - 100kHz Standard Mode, 1.7V to 3.6V
 - 400kHz Fast Mode, 1.7V to 3.6V
 - 1MHz Fast Mode Plus (FM+), 2.5V to 3.6V
- Schmitt Triggers, Filtered Inputs for Noise Suppression
- Bidirectional Data Transfer Protocol
- Write Protect Pin for Full Array Hardware Data Protection
- Ultra Low Active Current (1mA max) and Standby Current (0.8μA max)
- 8-byte Page Write Mode
 - Partial Page Writes Allowed
- Random and Sequential Read Modes
- Self-timed Write Cycle Within 5ms max
- High Reliability
 - Endurance: 1,000,000 Write Cycles
 - Data Retention: 100 Years
- Green Package Options (Lead-free/Halide-free/RoHS Compliant)
 - 8-lead SOIC, 8-lead TSSOP, 8-pad UDFN, 8-lead PDIP⁽¹⁾, 5-lead SOT23, and 8-ball VFBGA
- Die Sale Options: Wafer Form and Tape and Reel

Description

The Atmel® AT24C01D/02D provides 1,024/2,048 bits of Serial Electrically Erasable and Programmable Read-Only Memory (EEPROM) organized as 128/256 words of eight bits each. The device's cascadable feature allows up to eight devices to share a common 2-wire bus. These device are optimized for use in many industrial and commercial applications where low-power and low-voltage operation are essential. Both devices are available in space-saving 8-lead SOIC, 8-lead TSSOP, 8-pad UDFN, 8-lead PDIP⁽¹⁾, 5-lead SOT23, and 8-ball VFBGA packages. The entire family of packages operates from 1.7V to 3.6V.

Note: 1. Contact Atmel Sales for availability of this package.

B.4. Filtro EMI EClamp2398P



SEMTECH

PROTECTION PRODUCTS - EMIClamp®

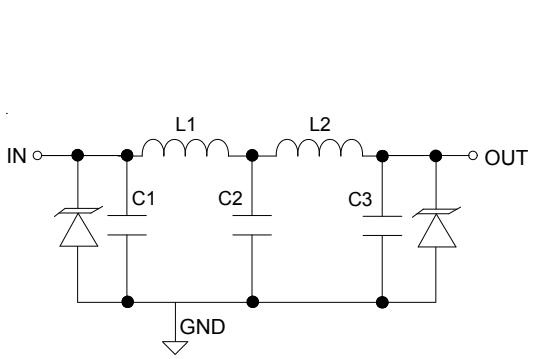
Description

EClamp®2398P is a low pass (L-C) filter array with integrated TVS diodes. It is designed to suppress unwanted EMI/RFI signals and provide electrostatic discharge (ESD) protection in portable electronic equipment. This state-of-the-art device utilizes solid-state silicon-avalanche technology for superior clamping performance and DC electrical characteristics. They have been optimized for **protection of color LCD and camera lines** in cellular phones and other portable electronics.

EClamp2398P device consists of eight identical circuits comprised of TVS diodes for ESD protection, and a 5-pole inductor - capacitor network for EMI/RFI filtering. A typical inductor value of 19nH and a capacitor value of 12pF are used to achieve 30dB minimum attenuation from 800MHz to 2.7GHz. The TVS diodes provide effective suppression of ESD voltages in excess of $\pm 15\text{kV}$ (air discharge) and $\pm 8\text{kV}$ (contact discharge) per IEC 61000-4-2, level 4.

EClamp2398P is in a 16-pin SLP4016P16 package measuring 4.0 x 1.6 x 0.58mm. Leads are spaced at a pitch of 0.5mm and are finished with lead-free NiPd. The small package makes it ideal for use in portable electronics such as cell phones, digital still cameras, and PDAs.

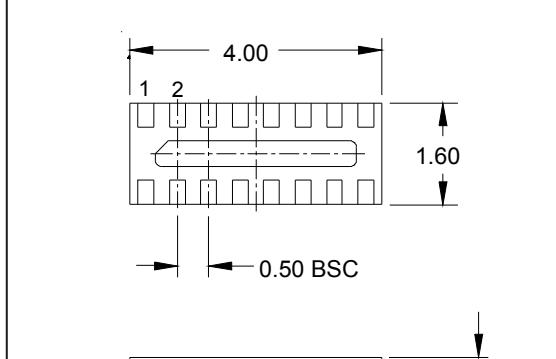
Circuit Diagram (Each Line)



Device Schematic (8X)

The schematic shows a single line of the filter array. It consists of an input terminal (IN), followed by a series inductor L1, a shunt capacitor C1, a TVS diode, and a ground connection. This is followed by another inductor L2, a shunt capacitor C2, a TVS diode, and a ground connection. Finally, there is a third inductor L3, a shunt capacitor C3, a TVS diode, and an output terminal (OUT).

Package Configuration



16 Pin SLP package (Bottom Side View)
Nominal Dimensions in mm

The package configuration diagram shows the physical dimensions of the 16-pin SLP package. The total width is 4.00 mm, the height is 1.60 mm, and the lead pitch (BSC) is 0.50 mm. The thickness of the package is 0.58 mm.

EClamp2398P
ESD/EMI Protection
for Color LCD Interfaces

Features

- ◆ Bidirectional EMI/RFI filter with integrated TVS for ESD protection
- ◆ ESD protection to **IEC 61000-4-2 (ESD) Level 4, $\pm 20\text{kV}$ (air), $\pm 15\text{kV}$ (contact)**
- ◆ Filter performance: 30dB minimum attenuation 800MHz to 2.7GHz
- ◆ TVS working voltage: 5V
- ◆ Inductor: 19nH (Typical)
- ◆ Capacitance: 12pF (Typical at VR = 2.5V)
- ◆ Protection and filtering for eight lines
- ◆ Solid-state technology

Mechanical Characteristics

- ◆ SLP4016P16 16-pin package
- ◆ Pb-Free, Halogen Free, RoHS/WEEE Compliant
- ◆ Nominal Dimensions: 4.0 x 1.6 x 0.58 mm
- ◆ Lead Pitch: 0.5mm
- ◆ Lead finish: NiPd
- ◆ Marking: Marking Code
- ◆ Packaging: Tape and Reel

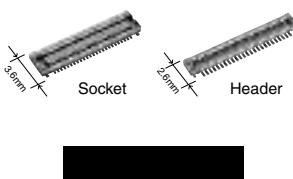
Applications

- ◆ Color LCD Protection
- ◆ Cell Phone CCD Camera Lines
- ◆ Clamshell Cell Phones

B.5. Conector sensor OV5680 - AXT630124

| | |
|--|-------------------|
| For board-to-FPC | F4S Series |
| Narrow pitch connectors (0.4mm pitch) | |



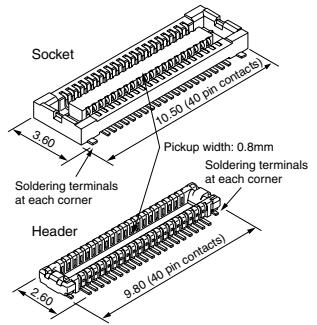


Socket
3.6mm

Header
2.6mm

FEATURES

- 1. Space-saving (3.6 mm widthwise)**
Smaller compared to F4 series (40 pin contacts):
Socket — 27% smaller,
Header — 38% smaller
- 2. Strong resistance to adverse environments!** Utilizes
“**TOUGH CONTACT**” construction
for high contact reliability.
- 3. Simple lock structure provides tactile feedback to ensure excellent mating/unmating operation feel.**
- 4. Gull-wing-shaped terminals to facilitate visual inspections.**
- 5. Connectors for inspection available**



Socket
3.6mm

Header
2.6mm

Pickup width: 0.8mm

Soldering terminals at each corner

10.50 (40 pin contacts)

Header
9.80 (40 pin contacts)

APPLICATIONS

Mobile devices, such as cellular phones, digital still cameras and digital video cameras.

ORDERING INFORMATION

| | | | | | |
|--|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|
| 5: Narrow Pitch Connector F4S (0.4 mm pitch) Socket | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | 2 | 4 |
| 6: Narrow Pitch Connector F4S (0.4 mm pitch) Header | <input type="checkbox"/> |
| Number of pins (2 digits) | | | | | |
| Mated height | | | | | |
| <Socket> | | | | | |
| 1: For mated height 1.0 mm | | | | | |
| 2: For mated height 1.2 mm | | | | | |
| <Header> | | | | | |
| 1: For mated height 1.0 mm | | | | | |
| 2: For mated height 1.2 mm | | | | | |
| Functions | | | | | |
| <Socket, Header> | | | | | |
| 2: Without positioning bosses | | | | | |
| Surface treatment (Contact portion / Terminal portion) | | | | | |
| <Socket> | | | | | |
| 4: Base: Ni plating Surface: Au plating (for Ni barrier available) | | | | | |
| <Header> | | | | | |
| 4: Base: Ni plating Surface: Au plating | | | | | |

Note: Please note that models with a mated height of 1.0 mm (7th digit of part number is "1") and 1.2 mm (7th digit of part number is "2") are not compatible.

Panasonic Corporation Automation Controls Business Division industrial.panasonic.com/ac/e/

ACCTB29E 201303-T

B.6. Chip Inductors

O05E.pdf
Apr.28,2017

muRata
INNOVATOR IN ELECTRONICS

Chip Inductors (Chip Coils)



B.7. Conector QSH-060-01-H-D-A

F-218 (Rev 10OCT17)

QSTRIP®

(0.50 mm) .0197"

HIGH-SPEED GROUND PLANE SOCKET

SPECIFICATIONS

For complete specifications and recommended PCB layouts see www.samtec.com/QSH

Insulator Material: Liquid Crystal Polymer
Contact Material: Phosphor Bronze
Plating: Au or Sn over 50 μ " (1.27 μ m) Ni
Current Rating: Contact: 2 A per pin (2 pins powered)
 Ground Plane: 25 A per ground plane (1 ground plane powered)
Operating Temp Range: -55 °C to +125 °C
Voltage Rating: 175 VAC (5 mm Stack Height)
Max Cycles: 100
RoHS Compliant: Yes

PROCESSING

Lead-Free Solderable: Yes
 SMT Lead Coplanarity: (0.10 mm) .004" max (020-060)
Board Stacking: For applications requiring more than two connectors per board contact ipg@samtec.com

RECOGNITIONS

For complete scope of recognitions see www.samtec.com/quality

PROTOCOLS

- 100 GbE
- Hypertransport™
- XAU
- PCI Express®
- InfiniBand™

ALSO AVAILABLE (MOQ Required)

- 15 mm, 22 mm and 30 mm stack height
- 30 μ " (0.76 μ m) Gold (Specify -H plating for Data Rate cable mating applications.)
- Edge Mount & Guide Posts
- 80 (-DP), 120, 150 positions per row
- Retention Option

Contact Samtec.

Board Mates: QTH

Cable Mates: HQCD, HQDP (See Also Available Note)

Standoffs: SO

Blade & Beam Design

EXTENDED LIFE PRODUCT
HIGH MATING CYCLES

QSH-060-01-L-D-DP-A
QSH-060-01-L-D-A-K
QSH-030-01-F-D-A

POWER/SIGNAL APPLICATION

Integral metal plane for power or ground

Compatible with UMP/TUMPS for flexible two-piece power/signal solutions

HIGH-SPEED CHANNEL PERFORMANCE

QTH/QSH @ 5 mm Mated Stack Height
Rating based on Samtec reference channel
For full SI performance data visit Samtec.com or contact SIG@samtec.com

25 Gb/s

QSH — **PINS PER ROW NO. OF PAIRS** — **01** — **PLATING OPTION** — **TYPE** — **A** — **OTHER OPTION**

-030, -060, -090
(60 total pins per bank = -D)
-020, -040, -060
(20 pairs per bank = -D-DP)

-F = Gold Flash on Signal Pins and Ground Plane, Matte Tin on tails
-L = 10 μ " (0.25 μ m) Gold on Signal Pins and Ground Plane, Matte Tin on tails
-C* = Electro-Polished Selective 50 μ " (1.27 μ m) min Au over 150 μ " (3.81 μ m) Ni on Signal Pins in contact area, 10 μ " (0.25 μ m) min Au over 50 μ " (1.27 μ m) Ni on Ground Plane in contact area, Matte Tin over 50 μ " (1.27 μ m) min Ni on all solder tails

-D = Single-Ended
-D-DP = Differential Pair (-01 only)
-K = (8.25 mm) .325" DIA Polyimide Film Pick & Place Pad
-TR = Tape & Reel (-090 positions maximum)
-L = Latching Option (Not available on -060 (-D-DP) & -090 positions)

QTH LEAD STYLE **MATED HEIGHT WITH QSH***

| | |
|-----|--------------|
| -01 | (5.00) .197 |
| -02 | (8.00) .315 |
| -03 | (11.00) .433 |
| -04 | (16.00) .630 |
| -05 | (19.00) .748 |
| -07 | (25.00) .984 |
| -09 | (14.00) .551 |

*Note: -C Plating passes 10 year MFG testing

Note: Some lengths, styles and options are non-standard, non-returnable.

Due to technical progress, all designs, specifications and components are subject to change without notice.

WWW.SAMTEC.COM

All parts within this catalog are built to Samtec's specifications.
 Customer specific requirements must be approved by Samtec and identified in a Samtec customer-specific drawing to apply.

B.8. TCA9548A

[Product Folder](#)
 [Sample & Buy](#)
 [Technical Documents](#)
 [Tools & Software](#)
 [Support & Community](#)

 **TEXAS INSTRUMENTS**

TCA9548A

SCPS207F – MAY 2012 – REVISED NOVEMBER 2016

TCA9548A Low-Voltage 8-Channel I²C Switch with Reset

1 Features

- 1-to-8 Bidirectional Translating Switches
- I²C Bus and SMBus Compatible
- Active-Low Reset Input
- Three Address Pins, Allowing up to Eight TCA9548A Devices on the I²C Bus
- Channel Selection Through an I²C Bus, In Any Combination
- Power Up With All Switch Channels Deselected
- Low R_{ON} Switches
- Allows Voltage-Level Translation Between 1.8-V, 2.5-V, 3.3-V, and 5-V Buses
- No Glitch on Power Up
- Supports Hot Insertion
- Low Standby Current
- Operating Power-Supply Voltage Range of 1.65 V to 5.5 V
- 5-V Tolerant Inputs
- 0- to 400-kHz Clock Frequency
- Latch-Up Performance Exceeds 100 mA Per JESD 78, Class II
- ESD Protection Exceeds JESD 22
 - ±2000-V Human-Body Model (A114-A)
 - 200-V Machine Model (A115-A)
 - ±1000-V Charged-Device Model (C101)

2 Applications

- Servers
- Routers (Telecom Switching Equipment)
- Factory Automation
- Products With I²C Slave Address Conflicts (Such as Multiple, Identical Temperature Sensors)

3 Description

The TCA9548A device has eight bidirectional translating switches that can be controlled through the I²C bus. The SCL/SDA upstream pair fans out to eight downstream pairs, or channels. Any individual SCn/SDn channel or combination of channels can be selected, determined by the contents of the programmable control register. These downstream channels can be used to resolve I²C slave address conflicts. For example, if eight identical digital temperature sensors are needed in the application, one sensor can be connected at each channel: 0-7.

The system master can reset the TCA9548A in the event of a time-out or other improper operation by asserting a low in the RESET input. Similarly, the power-on reset deselects all channels and initializes the I²C/SMBus state machine. Asserting RESET causes the same reset and initialization to occur without powering down the part. This allows recovery should one of the downstream I²C buses get stuck in a low state.

The pass gates of the switches are constructed so that the VCC pin can be used to limit the maximum high voltage, which is passed by the TCA9548A. Limiting the maximum high voltage allows the use of different bus voltages on each pair, so that 1.8-V, 2.5-V or 3.3-V parts can communicate with 5-V parts, without any additional protection. External pullup resistors pull the bus up to the desired voltage level for each channel. All I/O pins are 5-V tolerant.

Device Information⁽¹⁾

| PART NUMBER | PACKAGE | BODY SIZE (NOM) |
|-------------|------------|-------------------|
| TCA9548A | TSSOP (24) | 7.80 mm × 4.40 mm |
| | VQFN (24) | 4.00 mm × 4.00 mm |

(1) For all available packages, see the orderable addendum at the end of the data sheet.

Simplified Application Diagram

```

graph LR
    Master["I2C or SMBus Master  
(processor)"] -- SDA --> TCA9548A[ ]
    Master -- SCL --> TCA9548A
    Master -- RESET --> GND
    TCA9548A -- SD0 --> SlaveA0[Slaves A0, A1...An]
    TCA9548A -- SD1 --> SlaveB0[Slaves B0, B1...Bn]
    TCA9548A -- SD2 --> SlaveC0[Slaves C0, C1...Cn]
    TCA9548A -- SD3 --> SlaveH0[Slaves H0, H1...Hn]
    TCA9548A -- SD4 --> SlaveA1[ ]
    TCA9548A -- SD5 --> SlaveB1[ ]
    TCA9548A -- SD6 --> SlaveC1[ ]
    TCA9548A -- SD7 --> SlaveH1[ ]
    GND --- TCA9548A
  
```

An IMPORTANT NOTICE at the end of this data sheet addresses availability, warranty, changes, use in safety-critical applications, intellectual property matters and other important disclaimers. PRODUCTION DATA.

B.9. Cable Molex FPC 30 pin



This document was generated on 12/23/2014

PLEASE CHECK WWW.MOLEX.COM FOR LATEST PART INFORMATION

Part Number:

98266-0325

Status:

Active

Overview:

Premo-Flex™ FFC and Etched Polyimide Jumpers

Description:

0.50mm Pitch Premo-Flex™ FFC Jumper, Same Side Contacts (Type A), 30 Circuits, 0.152m Cable Length, 0.22mm Cable Thickness, Tin(Sn) Plating

Documents:

Drawing (PDF)

Product Specification PS-98266-001 (PDF)

[Packaging Specification PK-98266-001 \(PDF\)](#)

[RoHS Certificate of Compliance \(PDF\)](#)



image - Reference only

General

Product Family

Cable

Series

98266

Comments

Contacts on the same side, Type A

Crimp Quality Equipment

Yes

Overview

[Premo-Flex™ FFC and Etched Polyimide Jumpers](#)

Product Name

Promo-Flex™FFC Jumper

UPC

756054772812

Physical

Cable Length

0.152m

Circuits (Loaded)

30

Net Weight

1.481/g

Packaging Type

Bag

Pitch - Termination Interface

0.50mm

Temperature Range - Operating

-40°C to +105°C

Wire Size AWG

N/A

Wire/Cable Type

Flat Flex Cable

Electrical

Current - Maximum per Contact

0.5A

Voltage - Maximum

60V AC

Material Info

Reference - Drawing Numbers

Packaging Specification

PK-98266-001

Product Specification

PS-98266-001

Sales Drawing

SD-98266-001

EU RoHS

ELV and RoHS Compliant

REACH SVHC

**Contains SVHC: No
Low-Halogen Status
Not Low-Halogen**

China RoHS



Need more information on product environmental compliance?

Email productcompliance@molex.com

For a multiple part number RoHS Certificate of Compliance, [click here](#)

Please visit the Contact Us section for any non-product compliance questions.

Search Parts in this Series

98266Series

Use With

Molex FFC – FPC connector. Be sure to match like plating

This document was generated on 12/23/2014

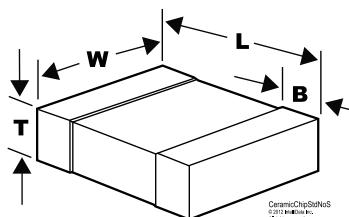
PLEASE CHECK WWW.MOLEX.COM FOR LATEST PART INFORMATION

B.10. C1206C120J5GACTU

KEMET Part Number: C1206C120J5GACTU
(C1206C120J5GAC7800)



Capacitor, Ceramic, SMD, MLCC, Ultra-Stable, Low Loss, Class I, 12 pF, +/-5% Tol, 50 V, COG, 1206 (3216 metric)



| General Information | |
|--------------------------|--|
| Supplier: | KEMET |
| Application: | Standard Chip |
| Chip Size: | 1206 (3216 metric) |
| Temperature Coefficient: | COG |
| Part Type Description: | SMD, MLCC, Ultra-Stable, Low Loss, Class I |
| Termination Type: | Tin (Sn) |
| Marked: | No |
| RoHS: | Yes |

| Dimensions (mm) | | |
|-----------------|-----------|-----------|
| Symbol | Dimension | Tolerance |
| L | 3.2 | +/-0.2 |
| W | 1.6 | +/-0.2 |
| T | 0.78 | +/-0.10 |
| B | 0.5 | +/-0.25 |

| Packaging Specifications | |
|--------------------------|--------------|
| Package Kind: | T&R |
| Package Size: | 7 in/180 mm |
| Package Type: | Plastic Tape |
| Package Quantity: | 4000 |

| Specifications | |
|------------------------|---|
| Capacitance: | 12 pF |
| Voltage: | 50 V |
| Tolerance: | +/-5% |
| Temperature Range: | -55/+125C |
| Dissipation Factor: | 0.1% |
| Failure Rate: | N/A |
| Aging Rate: | 0% loss/decade hour |
| Insulation Resistance: | 100 GOhm |
| Dielectric Strength: | 125 V |
| Miscellaneous: | Note: KEMET 50V rated parts may be operated at 63 Volts |

Statements of suitability for certain applications are based on our knowledge of typical operating conditions for such applications, but are not intended to constitute - and we specifically disclaim - any warranty concerning suitability for a specific customer application or use. This information is intended for use only by customers who have the requisite experience and capability to determine the correct products for their application. Any technical advice inferred from this information or otherwise provided by us with reference to the use of our products is given gratis, and we assume no obligation or liability for the advice given or results obtained.



© 2006-2016 IntelliData.net

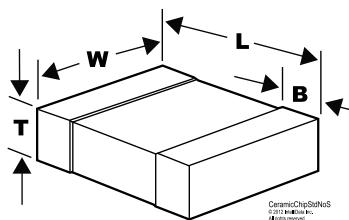
166ANEXO B. DATASHEETS DE LOS INTEGRADOS Y COMPONENTES UTILIZADOS

B.11. C1206C104J5RACTU

KEMET Part Number: C1206C104J5RACTU
(C1206C104J5RAC7800)



Capacitor, Ceramic, SMD, MLCC, Temperature Stable, Class II, 0.1 uF, +/-5% Tol, 50 V, X7R, 1206 (3216 metric)



| General Information | |
|--------------------------|---|
| Supplier: | KEMET |
| Application: | Standard Chip |
| Chip Size: | 1206 (3216 metric) |
| Temperature Coefficient: | X7R |
| Part Type Description: | SMD, MLCC, Temperature Stable, Class II |
| Termination Type: | Tin (Sn) |
| Marked: | No |
| RoHS: | Yes |

| Dimensions (mm) | | |
|-----------------|-----------|-----------|
| Symbol | Dimension | Tolerance |
| L | 3.2 | +/-0.2 |
| W | 1.6 | +/-0.2 |
| T | 0.78 | +/-0.10 |
| B | 0.5 | +/-0.25 |

| Packaging Specifications | |
|--------------------------|--------------|
| Package Kind: | T&R |
| Package Size: | 7 in/180 mm |
| Package Type: | Plastic Tape |
| Package Quantity: | 4000 |

| Specifications | |
|------------------------|--|
| Capacitance: | 0.1 uF |
| Voltage: | 50 V |
| Tolerance: | +/-5% |
| Temperature Range: | -55/+125C |
| Dissipation Factor: | 2.5% |
| Failure Rate: | N/A |
| Aging Rate: | 3% loss/decade hour |
| Insulation Resistance: | 10 GOhm |
| Dielectric Strength: | 125 V |
| Miscellaneous: | Note: Referee time for X7R dielectric for this part number is 1000 hours |

Statements of suitability for certain applications are based on our knowledge of typical operating conditions for such applications, but are not intended to constitute - and we specifically disclaim - any warranty concerning suitability for a specific customer application or use. This information is intended for use only by customers who have the requisite experience and capability to determine the correct products for their application. Any technical advice inferred from this information or otherwise provided by us with reference to the use of our products is given gratis, and we assume no obligation or liability for the advice given or results obtained.

B.12. Multilayer Ceramic Chip Capacitors



August 2016

MULTILAYER CERAMIC CHIP CAPACITORS



C Series Commercial Grade General (Up to 50V)

Type:

- C0402 [EIA CC01005]
- C0603 [EIA CC0201]
- C1005 [EIA CC0402]
- C1608 [EIA CC0603]
- C2012 [EIA CC0805]
- C3216 [EIA CC1206]
- C3225 [EIA CC1210]
- C4532 [EIA CC1812]
- C5750 [EIA CC2220]

B.13. Think film chip Resistors



Thick Film Chip Resistors

Type CRG Series



Key Features

- Thick film resistors with a high power to size ratio, ideally suited to industrial and general purpose use.
- A range from 1 ohm to 10M and tolerances of 1% and 5%. Also including zero ohm links.
- Suitable for most applications, including high frequency operation, owing to the short lead structure and low capacitance.
- Seven Package Sizes
- Terminal finish: Matte Sn
- MSL Level 2

Precious metal terminations are screen printed onto a ceramic base and fired. The resistive element is screen printed and fired and the passivation layer added. Each resistor is trimmed to tolerance by laser. The pre-scribed tile is broken into strips, the end plating is fired on and the strips broken into individual components. Final termination is made by electroplating.

Characteristics – Electrical

| | | 0201 | | | 0402 | | | 0603 | | | 0805 | | | | |
|----------------------------|-----|------|------|------|-------|------|------|------|------|------|-------|------|------|------|------|
| Rated Power @ 70 °C (W) | | 0.05 | | | 0.063 | | | 0.1 | | | 0.125 | | | | |
| Resistance Range (Ohms) | Min | 10 | 1 | 11 | 10 | 1 | 11 | 1 | 101 | 1 | 11 | 1 | 101 | 1 | 11 |
| | Max | 1M0 | 10 | 1M0 | 2M0 | 10 | 3M3 | 100 | 1M0 | 10 | 10M | 100 | 1M0 | 10 | 10M |
| Tolerance (%) | | 1 | 5 | 5 | 1 | 5 | 5 | 1 | 1 | 5 | 5 | 1 | 1 | 5 | 5 |
| Code letter | | F | J | J | F | J | J | F | F | J | J | F | F | J | J |
| Selection Series | | E24 | E24 | E24 | E24 | E24 | E24 | E24 | E24 | E24 | E24 | E24 | E24 | E24 | E24 |
| | | E96 | | | E96 | | | E96 | | | E96 | | E96 | | |
| Temp. Coefficient (ppm/°C) | | ±200 | ±400 | ±200 | ±100 | ±400 | ±200 | ±200 | ±100 | ±200 | ±200 | ±200 | ±100 | ±400 | ±200 |

| | | 1206 | | | 2010 | | | 2512 | | | | | |
|----------------------------|-----|------|------|------|------|------|------|------|------|------|------|------|------|
| Rated Power @ 70 °C (W) | | 0.25 | | | 0.5 | | | 1 | | | | | |
| Resistance Range (Ohms) | Min | 1 | 101 | 1 | 11 | 1 | 101 | 1 | 11 | 1 | 101 | 1 | 11 |
| | Max | 100 | 1M0 | 10 | 10M | 100 | 1M0 | 10 | 10M | 100 | 1M0 | 10 | 10M |
| Tolerance (%) | | 1 | 1 | 5 | 5 | 1 | 1 | 5 | 5 | 1 | 1 | 5 | 5 |
| Code letter | | F | F | J | J | F | F | J | J | F | F | J | J |
| Selection Series | | E24 |
| | | E96 | | | | E96 | | | E96 | | E96 | | |
| Temp. Coefficient (ppm/°C) | | ±200 | ±100 | ±400 | ±200 | ±200 | ±100 | ±400 | ±200 | ±200 | ±100 | ±400 | ±200 |

| | | 0201 | | 0402 | | 0603 | | 0805 | | 1206 | | 2010 | | 2512 | |
|---------------------------------------|--|-------------|-----|------|-----|------|-----|------|-----|------|-----|------|-----|------|--|
| Working Voltage (V) | | 25 | 50 | 50 | 50 | 150 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | |
| Max. Overload Voltage (V) | | 50 | 100 | 100 | 100 | 300 | 400 | 400 | 400 | 400 | 400 | 400 | 400 | | |
| Operating Temp. Range (°C) | | -55 to +125 | | | | | | | | | | | | | |
| Climatic Category (°C) | | 55/125/56 | | | | | | | | | | | | | |
| Insulation Resistance Dry Min (Mohms) | | 1000 | | | | | | | | | | | | | |
| Stability (%) | | 3 | | | | | | | | | | | | | |
| Zerohm (A) Current Max | | 0.5 | 1 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | | |
| Resistance Max | | <50 mOhm | | | | | | | | | | | | | |

1773204 CIS WR 03/2012

Dimensions are in millimeters and inches unless otherwise specified. Values in brackets are standard equivalents.

Dimensions are shown for reference purposes only. Specifications subject to change.

For email, phone or live chat, go to: te.com/help

B.14. 1SMA5917BT3G

1SMA59xxBT3G Series, SZ1SMA59xxBT3G Series

1.5 Watt Plastic Surface Mount Zener Voltage Regulators

This complete new line of 1.5 Watt Zener Diodes offers the following advantages.

Features

- Standard Zener Breakdown Voltage Range – 3.3 V to 68 V
- ESD Rating of Class 3 (> 16 kV) per Human Body Model
- Flat Handling Surface for Accurate Placement
- Package Design for Top Slide or Bottom Circuit Board Mounting
- Low Profile Package
- Ideal Replacement for MELF Packages
- AEC-Q101 Qualified and PPAP Capable – SZ1SMA59xxBT3G
- SZ Prefix for Automotive and Other Applications Requiring Unique Site and Control Change Requirements
- These are Pb-Free Devices*

Mechanical Characteristics:

CASE: Void-free, transfer-molded plastic

FINISH: All external surfaces are corrosion resistant with readily solderable leads

MAXIMUM CASE TEMPERATURE FOR SOLDERING PURPOSES:

260°C for 10 seconds

POLARITY: Cathode indicated by molded polarity notch or cathode band

FLAMMABILITY RATING: UL 94 V-0 @ 0.125 in

MAXIMUM RATINGS

| Rating | Symbol | Value | Unit |
|--|----------------|-------------------|--|
| DC Power Dissipation @ $T_L = 75^\circ\text{C}$, Measured Zero Lead Length (Note 1) Derate above 75°C Thermal Resistance, Junction-to-Lead | P_D | 1.5 20 50 | W mW/ $^\circ\text{C}$ $^\circ\text{C}/\text{W}$ |
| | $R_{0,UL}$ | | |
| DC Power Dissipation @ $T_A = 25^\circ\text{C}$ (Note 2) Derate above 25°C Thermal Resistance, Junction-to-Ambient | P_D | 0.5 4.0 250 | W mW/ $^\circ\text{C}$ $^\circ\text{C}/\text{W}$ |
| | $R_{0,JA}$ | | |
| Operating and Storage Temperature Range | T_J, T_{Stg} | -65 to +150 | $^\circ\text{C}$ |

Stresses exceeding those listed in the Maximum Ratings table may damage the device. If any of these limits are exceeded, device functionality should not be assumed, damage may occur and reliability may be affected.

1. 1 in square copper pad, FR-4 board.
2. FR-4 Board, using ON Semiconductor minimum recommended footprint.

*For additional information on our Pb-Free strategy and soldering details, please download the ON Semiconductor Soldering and Mounting Techniques Reference Manual, SOLDERRM/D.

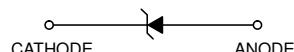


ON Semiconductor®

www.onsemi.com



SMA
CASE 403D
STYLE 1



MARKING DIAGRAM



8xxB = Device Code (Refer to page 2)
A = Assembly Location
Y = Year
WW = Work Week
■ = Pb-Free Package

ORDERING INFORMATION

| Device | Package | Shipping† |
|----------------|------------------|------------------------|
| 1SMA59xxBT3G | SMA (Pb-Free) | 5,000 / Tape & Reel |
| SZ1SMA59xxBT3G | SMA (Pb-Free) | 5,000 / Tape & Reel |

†For information on tape and reel specifications, including part orientation and tape sizes, please refer to our Tape and Reel Packaging Specifications Brochure, BRD8011/D.

DEVICE MARKING INFORMATION

See specific marking information in the device marking column of the Electrical Characteristics table on page 2 of this data sheet.

B.15. Conector de pines

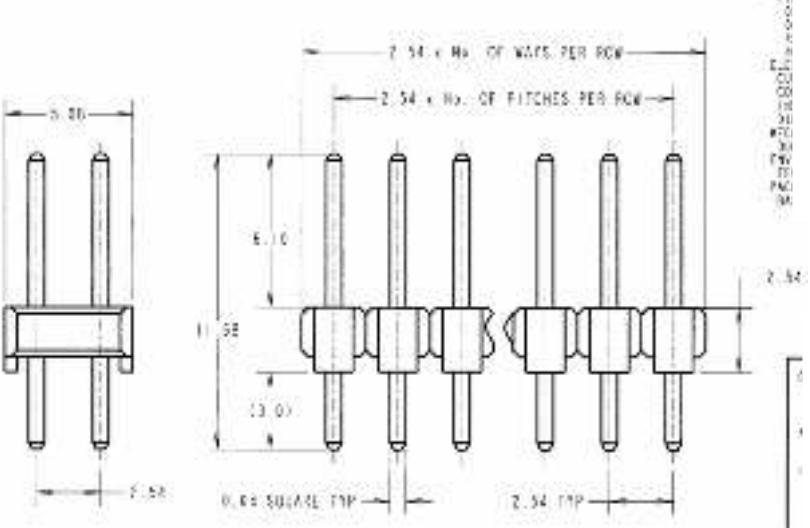


ENGLISH

Datasheet

Straight PCB Header
RS Stock number [547 - 3166](#)

Dimensions: (mm)



Specification

- Recommended PCB mounting hole size = Ø1.00mm
- All tolerances are ± 0.25 mm unless otherwise stated

| | |
|----------------------|--|
| Material | <ul style="list-style-type: none">Moulding: 0.5 & 0.6 = Glass- Filled Polyester UL94V-0, Black45 & 46 = High temp Nylon UL94V-0, BlackPins: BrassFinish:<ul style="list-style-type: none">05 = gold06 = Tin/Lead45 = Gold46 = 100% Tin |
| Electrical | Current Rating: 3A Max Contact Resistance: 25m Ω Max (gold), 50 operations (tin) Dielectric Strength: 750V AC/DC |
| Mechanical | Durability: 300 operations (gold) 50 operations (tin) |
| Environmental | Temperature Range: -40°C to +105 °C |

RS, Professionally Approved Products, gives you professional quality parts across all products categories. Our range has been testified by engineers as giving comparable quality to that of the leading brands without paying a premium price.

Anexo C

Manuales de referencia de la NVIDIA JETSON TX2

En este anexo, se presentan los diferentes manuales de referencia de la **NVIDIA JETSON TX2** consultados durante el desarrollo de este proyecto. Además se añaden manuales de la anterior, versión de la placa de desarrollo. Por otro lado, se adjunta el manual de referencia del conector *J20* desarrollado por **Auvidea**.

C.1. Developer Kit Carrier Board Specification



SPECIFICATION
NVIDIA Jetson TX1/TX2 Developer Kit
Carrier Board

Abstract

This document contains recommendations and guidelines for Engineers to follow to create modules for the expansion connectors on the Jetson™ carrier board as well as understand the capabilities of the other dedicated interface connectors and associated power solutions on the platform. The Jetson TX1/TX2 Developer Kit Carrier Board (P2597) supports Jetson TX1, Jetson TX2 & Jetson TX2i.

Note: Jetson TX2/TX2i utilizes Tegra X2 which is a Parker series SoC.

- CAUTION:**
1. ALWAYS CONNECT JETSON MODULE & ALL EXTERNAL PERIPHERAL DEVICES BEFORE CONNECTING THE POWER SUPPLY TO THE AC POWER JACK. Connecting a device while powered on may damage the Developer Kit carrier board, Jetson module or peripheral device. In addition, the carrier board should be powered down and the power removed before plugging or unplugging devices or add-on modules into the headers. Wait for the red power VDD_IN LED (See Figure 1) to turn off, or wait for 5 minutes if your system does not have a power LED. This includes the Jetson module, the camera & display headers, the M.2 connector, the PCIe® x4 connector, SATA & the other expansion headers. For the PCIe4 & SATA connector, also wait for the PCIe/SATA 12V LED to turn off (See Figure 1)
 2. The NVIDIA® Jetson Developer Kit carrier board contains ESD-sensitive parts. Always use appropriate anti-static and grounding techniques when working with the system. Failure to do so can result in ESD discharge to sensitive pins, and irreparably damage your Jetson carrier board. NVIDIA will not replace units that have been damaged due to ESD discharge.

C.2. OEM Product Design Guide



OEM PRODUCT DESIGN GUIDE NVIDIA Jetson TX2/TX2i

Abstract

This document contains recommendations and guidelines for Engineers to follow to create a product that is optimized to achieve the best performance from the common interfaces supported by the NVIDIA® Jetson™ TX2/TX2i System-on-Module (SOM).

Notes: Jetson TX2 & Jetson TX2i modules utilize Tegra X2 which is a Parker series SoC.

C.3. Module DataSheet



DATA SHEET

NVIDIA Jetson TX2 / TX2i System-on-Module Pascal GPU + ARMv8 + 8GB LPDDR4 + 32GB eMMC

NVIDIA Jetson Modules: TX2, TX2i

References to TX2 include (can be read as) TX2i except where explicitly noted.

Description

The NVIDIA® Jetson TX2 series System-on-Module (SOM) redefines possibility; a combination of performance, power efficiency, integrated deep learning capabilities and rich I/O remove the barriers to a new generation of products. The Jetson TX2 is ideal for many applications including (but not limited to): Intelligent Video Analytics (IVA), Drones, Robotics, Gaming Devices, Virtual Reality (VR), Augmented Reality (AR) and Portable Medical Devices. Superior performance, robust design and reduced complexity in system integration results in more advanced products getting to market faster.

The Jetson TX2 series module integrates:

- **256 core NVIDIA Pascal GPU.** Fully supports all modern graphics APIs, unified shaders and is GPU compute capable. The GPU supports all the same features as discrete NVIDIA GPUs, including extensive compute APIs and libraries including CUDA. Highly power optimized for best performance in embedded use cases.
- **ARMv8 (64-bit) Multi-Processor CPU Complex.** Two CPU clusters connected by a high-performance coherent interconnect fabric designed by NVIDIA; enables simultaneous operation of both CPU clusters for a true heterogeneous multi-processing (HMP) environment. The **Denver 2 (Dual-Core)** CPU clusters is optimized for higher single-thread performance; the **ARM Cortex-A57 MPCore (Quad-Core)** CPU clusters is better suited for multi-threaded applications and lighter loads.
- **Advanced HD Video Encoder.** Recording of 4K ultra-high-definition video at 60fps. Supports H.265 and H.264 BP/MP/HP/MVC, VP9 and VP8 encoding.
- **Advanced HD Video Decoder.** Playback of 4K ultra-high-definition video at 60fps with up to 12-bit pixels. Supports H.265, H.264, VP9, VP8 VC-1, MPEG-2, and MPEG-4 video standards.
- **Display Controller Subsystem.** Two multi-mode (eDP/DP/HDMI) outputs and up to 8-lanes of MIPI-DSI output. Multiple line pixel storage allows more memory-efficient scaling operations and pixel fetching. Hardware display surface rotation is also provided for bandwidth reduction in mobile applications.
- **128-bit Memory Controller.** 128-bit DRAM interface providing high bandwidth LPDDR4 and ECC (**TX2i only**) support.
- **8GB LPDDR4 and 32 GB eMMC memory** integrated on the module
- **1.4Gpix/s Advanced image signal processing:** Hardware accelerated still-image and video capture path, with advanced ISP.
- **Audio Processing Engine.** Audio subsystem enables full hardware support for multi-channel audio over multiple interfaces.

C.4. OEM Product Design Guide



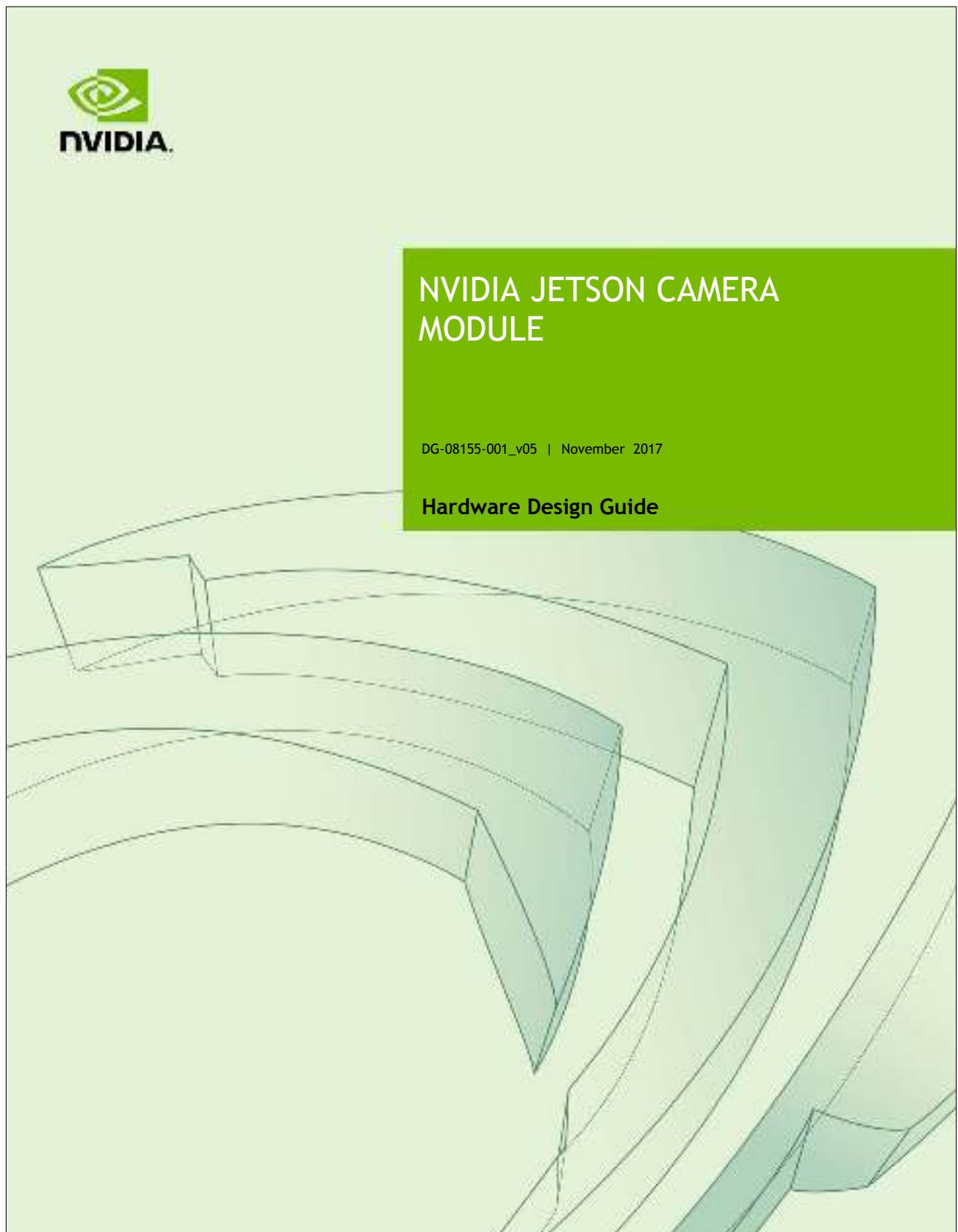
OEM PRODUCT DESIGN GUIDE NVIDIA Jetson TX1

Abstract

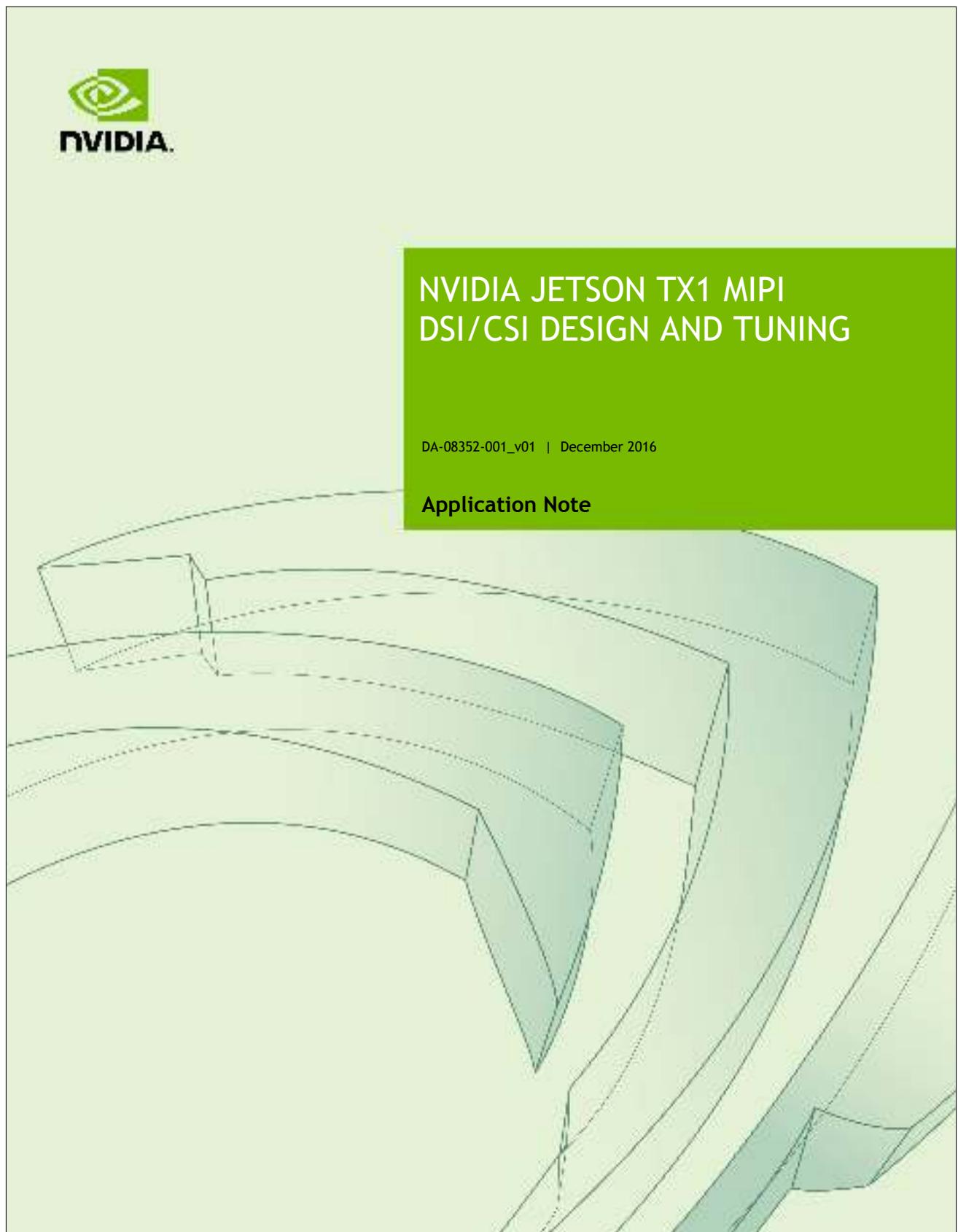
This document contains recommendations and guidelines for Engineers to follow to create a product that is optimized to achieve the best performance from the common interfaces supported by the NVIDIA® Jetson™ TX1 System-on-Module (SoM).

This document provides detailed information on the capabilities of the hardware module, which may differ from supported configurations by provided software. Refer to software release documentation for information on supported capabilities.

C.5. Camera Module Hardware Design Guide



C.6. MIPI DSI/CSI Design and Tuning



C.7. Manual de referencia del conector Auvidea J20

VERSION 1.0

J20 (38223)

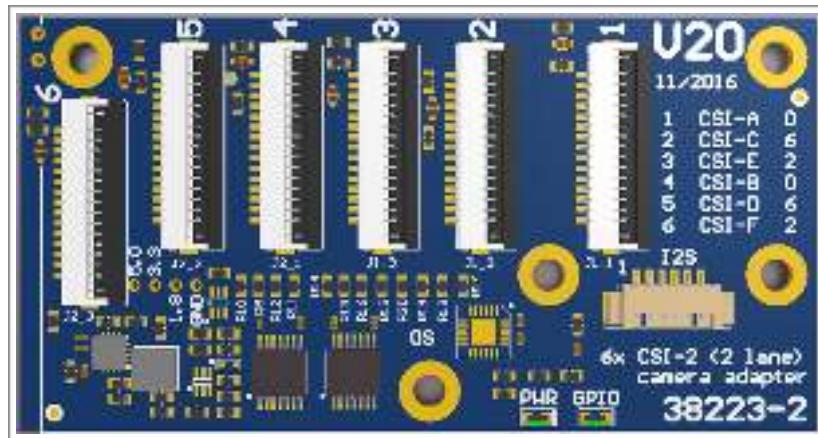


J20

technical reference manual

38223

Version 1.0



Preliminary

November 2016

Auvidea GmbH
Kellerberg 3
D-86920 Denklingen
Tel: +49 8243 7714 622
info@auvidea.com
www.auvidea.com

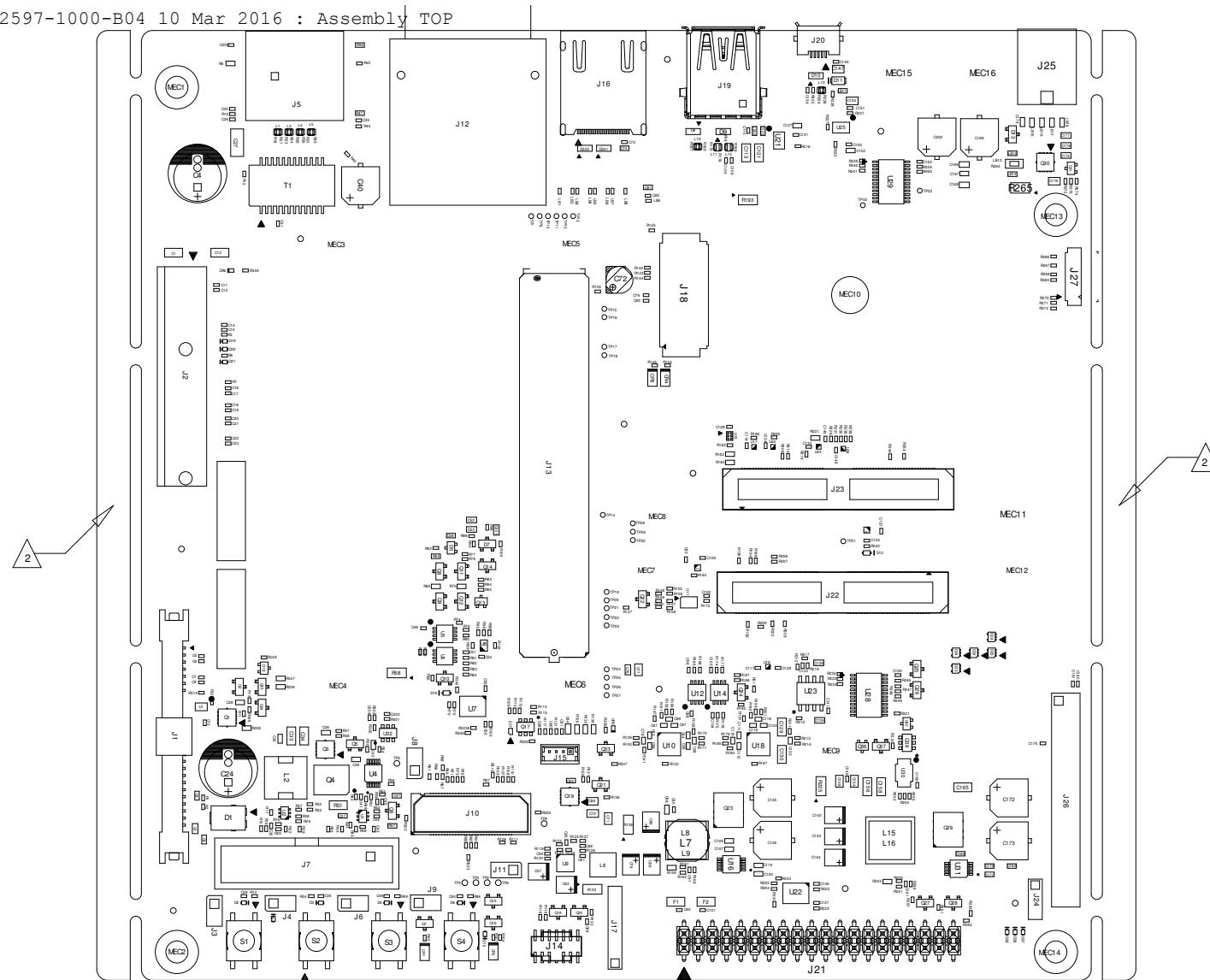
Anexo D

Hojas de características NVIDIA JETSON TX2

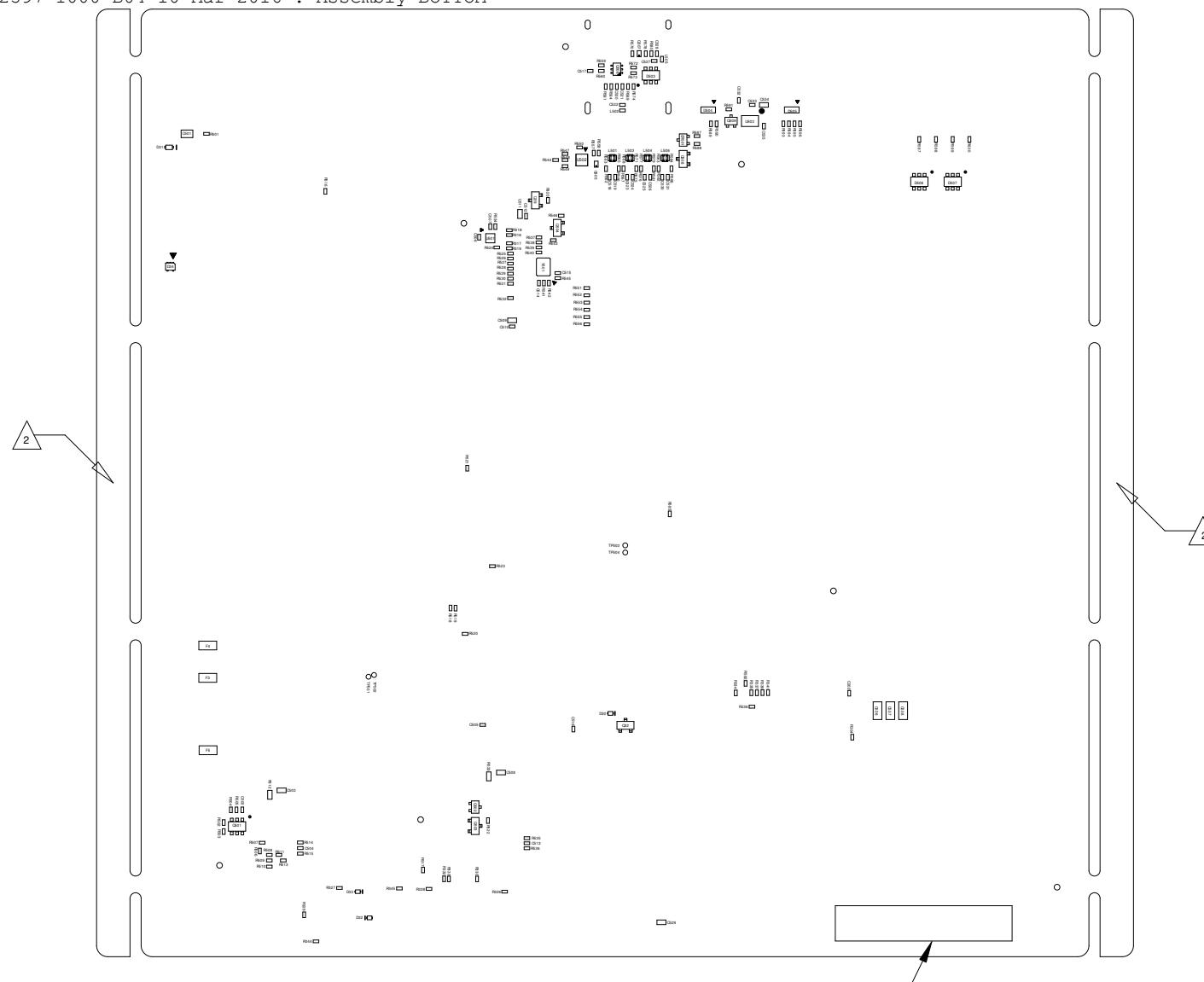
En este anexo, se presentan los planos consultados para conocer las especificaciones mecánicas de la **NVIDIA JETSON TX2**. Por otro lado, se presentan los diagramas de las especificaciones eléctricas del dispositivo y las especificaciones de los conectores usados durante el desarrollo del proyecto.

180 ANEXO D. HOJAS DE CARACTERÍSTICAS NVIDIA JETSON TX2

nvPLOT 142-82597-1000-B04 10 Mar 2016 : Assembly TOP



nvPLOT 142-82597-1000-B04 10 Mar 2016 : Assembly BOTTOM



DO NOT
SCALE FROM
THIS PRINT

| TABLE 1 | | |
|---------|-------------|--------------|
| | "A" | "B" |
| 030 | .085 [2.16] | .738 [19.18] |
| 060 | .162 [4.11] | .760 [19.30] |
| 090 | .237 [6.01] | .770 [19.53] |
| -120 | .304 [7.71] | .785 [19.91] |
| -150 | .360 [9.14] | .803 [20.34] |

| TABLE 1 | |
|-----------|------------------------------------|
| No OF POS | OPTIONALITY RETENTION GROUND PLANE |
| -260 | -C126-11-X |
| -290 | -C126-11-X |

QSH-XXX-01-X-D-XXX

NO OF POSITIONS
-030, -060, -090,
-*120, *-150 (PER ROW)
(* = SEE NOTE 11)

LEAD STYLE
-01, .1280[3.25]

OPTION
-A: ALIGNMENT PINS
(USE QSH-30-01-D-XX-A)
(NOT AVAILABLE WITH -LC)
-LC: LOCKING CLIPS
(USE LC-08-TM-01), SEE FIG 2, SHT 2
(NOT AVAILABLE WITH -A)
-SEENOTE 13)

-GP: GUIDE POSTS
(SEE FIG 4, SHT 3)
-TAPE & REEL FOR -GP, -060 POS MAX)

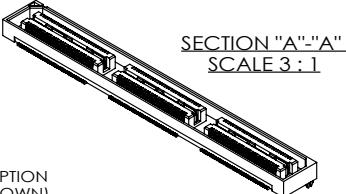
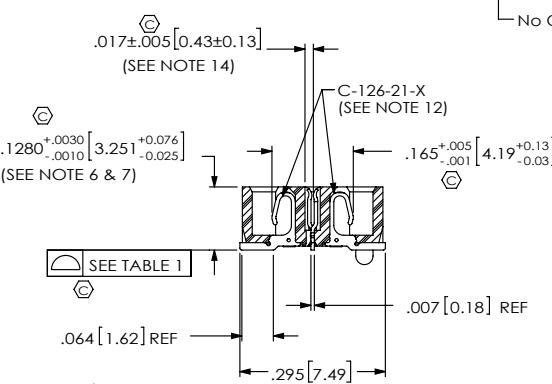
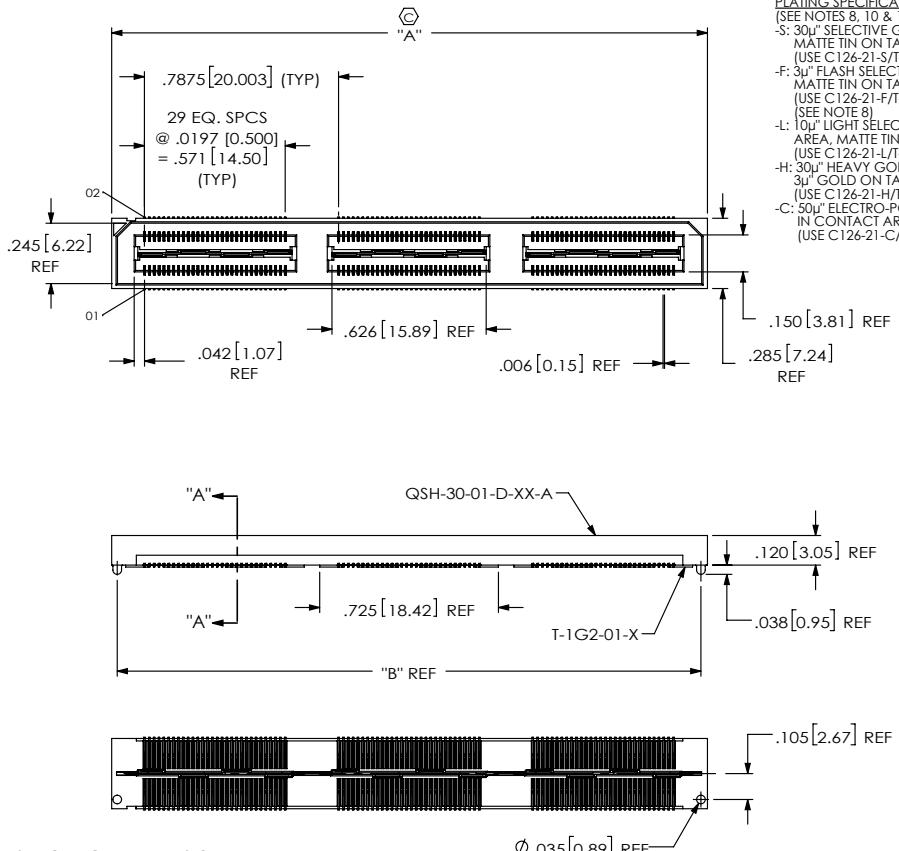
-K: POLYIMIDE FILM PAD
(USE K-DOT-325-.450-.005, SEE FIG 3, SHT 2)
-TR: TAPE & REEL (-020 POS MAX)
-FL: FRICTION LOCKING
(SEE FIG 5, SHT 3)

(NOT AVAILABLE WITH -GP)
-RTI: RETENTION (MUST USE -A OPTION)
(USE QSH-30-01-D-XX-A-RTI)
& SUB-SC-05-01.FF, SEE FIG 6, SHT 4)
(NOT AVAILABLE WITH -LC, -GP, OR -FL)
(NOT AVAILABLE WITH -120 & -150 POS)

-L: LATCHING
(USE QSH-30-01-D-XX-L, SEE FIG 7, SHT 4)
(NOT AVAILABLE WITH -LC, -GP, -FL, OR -RTI)
(NOT AVAILABLE WITH -090, -120, & -150 POS)

ROW SPECIFICATION
-D: DOUBLE (USE QSH-30-01-D-XX)

No OF BANKS



NOTES:
1. (C) REPRESENTS A CRITICAL DIMENSION.

2. NOTE DELETED.

3. CONTACT RETENTION: 8 OZ. MINIMUM.

4. GROUND PLANE RETENTION: 8 OZ. MINIMUM.

5. BURR ALLOWANCE: .0015[0.038] MAX.

6. MAX VARIANCE: .002[0.05].

7. DIMENSION APPLIES TO CONTACT & GROUND PLANE.

8. -L PLATING CAN BE SUBSTITUTED FOR -F PLATING.

9. SUB-SC-05-01-XX RETENTION TO BE 1.0 LBF IN BODY.

10. MAY SUB C-126-01 FOR C-126-21.

11. FOR NEW APPLICATIONS REQUIRING THESE POSITIONS, PLEASE CONTACT SAMTEC INTERCONNECT PROCESSING GROUP.

12. FOR -120, -150 POSITIONS USE C-269-11-X, DO NOT SUB C-126.

13. DUE TO THE HIGH AMOUNT OF INSERTION FORCE NEEDED, THE -LC OPTION IS NOT COMPATIBLE WITH AUTO PLACEMENT. SAMTEC RECOMMENDS MANUAL PLACEMENT FOR ALL ASSEMBLIES WITH THE -LC OPTION.

14. CHECK ONCE PER BANK ALTERNATING SIDES FOR MULTIPLE BANK PARTS.
15. SIDE WALLS ARE RIBBED AS NECESSARY TO CONTROL BOW.

FIG 1
-A: ALIGNMENT PIN OPTION
(QSH-090-01-X-D-A SHOWN)

UNLESS OTHERWISE SPECIFIED,
DIMENSIONS ARE IN INCHES.
TOLERANCES ARE:
DECIMALS ANGLES
XX: ±.01[.3] 2°
XXX: ±.005[.13]
XXXX: ±.0020[.051]

PROPRIETARY NOTE
* THIS DOCUMENT CONTAINS CONFIDENTIAL AND PROPRIETARY INFORMATION AND ALL DESIGN, MANUFACTURING, RELATED AND OTHER USE, PATH RIGHTS AND LICENSES HEREIN ARE EXPRESSLY RESERVED BY SAMTEC, INC. THIS DOCUMENT SHALL NOT BE DISCLOSED, IN WHOLE OR PART, TO ANY UNAUTHORIZED PERSON OR ENTITY NOR REPRODUCED, TRANSFERRED OR INCORPORATED IN ANY OTHER PROJECT IN ANY MANNER WITHOUT THE EXPRESS WRITTEN CONSENT OF SAMTEC, INC.

MATERIAL: DO NOT SCALE DRAWING SHEET SCALE: 3:2

INSULATOR: LCP UL94V-0, COLOR: BLACK
CONTACT & GROUND PLANE: PHOS BRONZE

F:\DWG\MISC\MKTG\QSH-XXX-01-X-D-XXX-MKT.SLDDRW

samtec

520 PARK EAST BLVD, NEW ALBANY, IN 47150
PHONE: 812-944-6733 FAX: 812-948-5047
e-Mail: info@SAMTEC.com code 55322

DESCRIPTION: .5mm DOUBLE ROW HS SOCKET ASSEMBLY

DWG. NO. QSH-XXX-01-X-D-XXX

BY: DEAN P 05/21/1999 SHEET 1 OF 4

D.1. Especificaciones eléctricas

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| A | B | C | D | E | F | G | H |
|---|---|---|---|---|---|---|---|

P2597-B04

Jetson TX1/TX2 Developer Kit Carrier Board
HDMI, USB, PCIe, SATA, Gbe

| | |
|-------------------|--------------------------------|
| TABLE OF CONTENTS | |
| Page | Description |
| 1 | Table of Contents |
| 2 | Block Diagram |
| 3 | Power Tree |
| 4 | IO connector 1/3 |
| 5 | IO connector 2/3 |
| 6 | IO connector 3/3 |
| 7 | BLANK |
| 8 | USB3.0 type A and USB2.0 micro |
| 9 | PCIe x4 Connector |
| 10 | M.2 KEY E |
| 11 | SDcard Slot |
| 12 | SATA |
| 13 | Gigabit Ethernet |
| 14 | HDMI |
| 15 | Serial port and JTAG |
| 16 | IO Expansion Connector |
| 17 | DSI Connector |
| 18 | CSI Connector |
| 19 | BLANK |
| 20 | BLANK |
| 21 | GPIO Expanders |
| 22 | FAN AND DEBUG |
| 23 | 12V BOOST |
| 24 | EEPROM |
| 25 | Supervisor |

| | |
|------|------------------------|
| Page | Description |
| 26 | BLANK |
| 27 | DC Jack |
| 28 | 5V & 3V3 Vregs |
| 29 | 3V3 SLP AND 1.8V Vregs |
| 30 | Rail Discharge |
| 31 | Power Monitor I |
| 32 | Power Monitor II |
| 33 | Buttons and Mechanical |
| 34 | REVISION HISTORY |

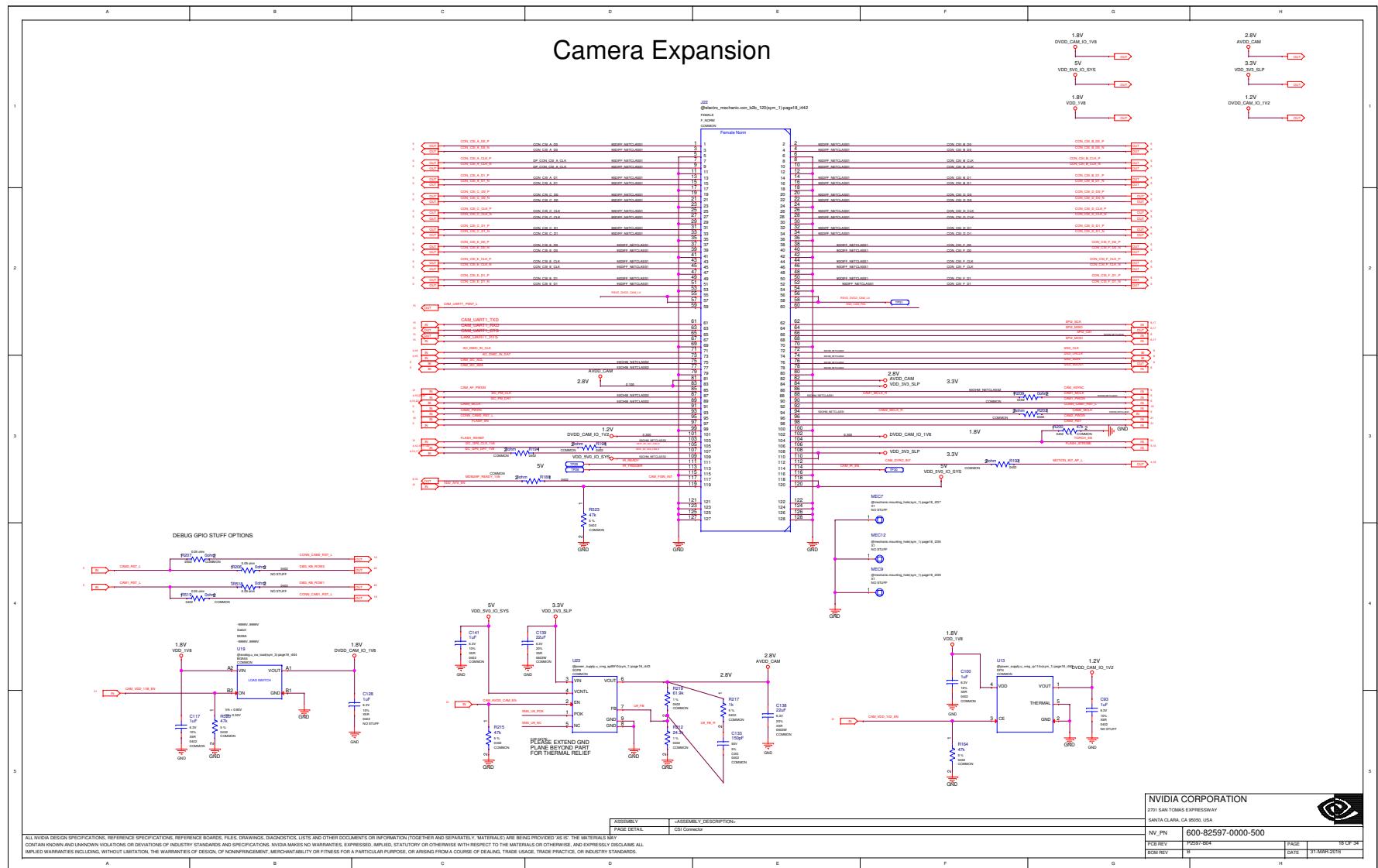
| | |
|----------------------------|--------------------|
| P2597 I2C Port Assignments | |
| Slave Device | BUS Device address |
| INA220 (Power measurement) | I2C Gen 2 : 7 h42 |
| INA221 (Power measurement) | I2C Gen 2 : 7 h43 |
| TCAS939 (GPIO Expander) | I2C Gen 1 : 7 h74 |
| TCAS939 (GPIO Expander) | I2C Gen 2 : 7 h77 |
| Si4732(E) EEPROM | I2C Gen 3 : 7 h79 |

| | |
|----------------------------|--------------------|
| P2140 I2C Port Assignments | |
| Slave Device | BUS Device address |
| TP4351 (Power control) | I2C Gen 1 : 7 h44 |
| ADS1115 (A/C detection) | I2C Gen 2 : 7 h44 |
| INA228 (Power measurement) | I2C Gen 2 : 7 h45 |
| INA221 (Power measurement) | I2C Gen 2 : 7 h46 |

| | |
|---------------------------|--------------------|
| NVIDIA CORPORATION | |
| 2701 SAN JOSE EXPRESSWAY | |
| SANTA CLARA, CA 95051 USA | |
| NV_PN | 600-82597-0000-500 |
| PCB REV | 17307-804 |
| BOM REV | 1 |
| PAGE | 1 OF 34 |
| DATE | 31-MAR-2016 |

NVIDIA DESIGN SPECIFICATIONS, REFERENCE SPECIFICATIONS, REFERENCE BOARDS, CIRCUITS, DRAWINGS, DIAGNOSTICS, LISTS AND OTHER DOCUMENTS OR INFORMATION (TOGETHER AND SEPARATELY, "MATERIALS") ARE BEING PROVIDED AS IS. THE MATERIALS DO NOT CONTAIN KNOWN AND UNKNOWN VIOLATIONS OR DEVIATIONS OF INDUSTRY STANDARDS AND SPECIFICATIONS. NVIDIA MAKES NO WARRANTIES, EXPRESSED, IMPLIED, STATUTORY OR OTHERWISE WITH RESPECT TO THE MATERIALS OR OTHERWISE, AND EXPRESSLY DISCLAIMS ALL IMPLIED WARRANTIES INCLUDING, WITHOUT LIMITATION, THE WARRANTIES OF DESIGN, NONINFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE, OR ARISING FROM A COURSE OF DEALING, TRADE USAGE, TRADE PRACTICE, OR INDUSTRY STANDARDS.

Camera Expansion



Anexo E

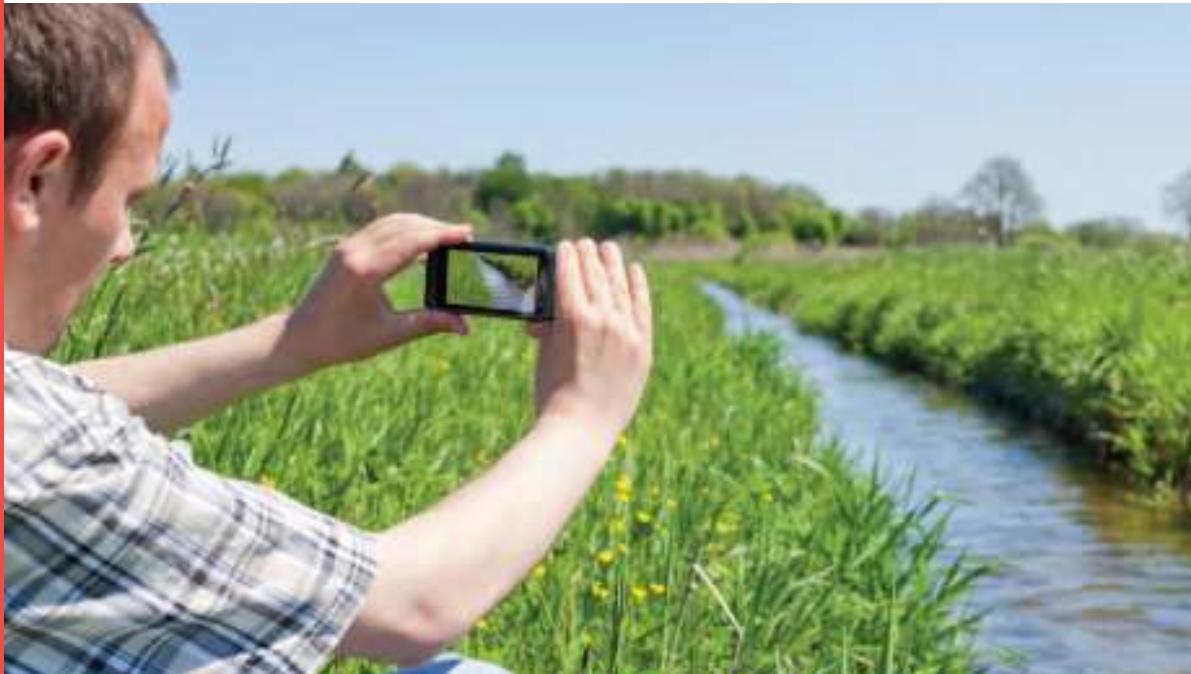
Hojas de características de sensores CMOS

En este anexo se presentan las diferentes hojas de características de los **sensores CMOS** estudiados durante el desarrollo del proyecto.

E.1. OmniVision OV5680



OV5680 5-megapixel product brief



 available in a lead-free package

5-Megapixel Image Sensor with OmniBSI-2™ Technology for Video-Centric Smartphones

The 5-megapixel OV5680 features OmniVision's advanced 1.75-micron OmniBSI-2 pixel architecture, designed to further narrow the performance gap between smartphones and dedicated digital video cameras. The 1/3.2-inch CMOS image sensor offers best-in-class image quality while capturing 1080p high-definition (HD) video at 30 frames per second (fps) for mobile applications.

The OV5680 utilizes an integrated scaler to provide 1080p HD video capture at 30 fps for continuous shooting and shutterless designs without any lag. The scaler enables electronic image stabilization, while maintaining full field of view in 1080p HD video mode. The sensor's 2x2 binning functionality with post-binning re-sampling filter enables 720p video capture at 60 fps, minimizes spatial artifacts and removes image artifacts around edges, delivering clean and crisp color images for best-in-class HD video.

The OV5680 can synchronize exposure and frame for stereo cameras to meet 3D video capture requirements. The new 1.75-micron OmniBSI-2 pixel is built using a 300 nm copper process with 65 nm design rules, offering optimized die size, lower power consumption, and significant performance and image quality improvements over the first-generation OmniBSI™ pixel.

The OV5680 comes with a standard 2-lane MIPI serial output interface and fits into the industry standard 8.5 x 8.5 x 6 mm module size.

Find out more at www.ovt.com.



E.2. OmniVision OV5693



available in
a lead-free
package

High-Performance 5-Megapixel Image Sensor for Front-Facing Cameras in Flagship Smartphones and Tablets

The OV5693 is OmniVision's highest performance 1/4-inch, 5-megapixel image sensor, delivering DSC-quality imaging and low-light performance as well as full 1080p high-definition video recording at 30 frames per second (fps). Using OmniVision's proprietary 1.4-micron OmniBSI-2™ pixel architecture, the OV5693 provides best-in-class low-light performance and image quality in a slim camera module. This makes the OV5693 an ideal camera solution for slim flagship smartphones and tablets, providing exceptional 5-megapixel "selfie" images and high-quality 1080p video.

Leveraging OmniVision's second-generation industry-leading backside illumination pixel technology, the OV5693 offers full resolution 5-megapixel images at 30 fps, an integrated scaler, and 2x2 binning

functionality with re-sampling filter. The scaler enables electronic image stabilization, while maintaining full field-of-view in both 720p and 1080p HD video modes. The 2x2 binning functionality, which features a post-binning re-sampling filter, further increases the sensor's sensitivity, while minimizing spatial artifacts and removing image artifacts around edges to produce crisp, clean color images.

The sensor features a high-speed 2-lane MIPI interface running up to 900 Mbps per lane and fits into an industry standard module size of 8.5 x 8.5 mm with a z-height of 4.2 mm for an autofocus module.

Find out more at www.ovt.com.



OmniVision.

E.3. OmniVision OV5647



E.4. OmniVision OV5640



E.5. Sony IMX219

SONY

Diagonal 4.60 mm (Type 1/4.0) 8 Mega-Pixel CMOS Image Sensor with Square Pixel for Color Cameras

IMX219PQH5-C

Description

The IMX219PQH5-C is a diagonal 4.60 mm (Type 1/4.0) CMOS active pixel type image sensor with a square pixel array and 8.08M effective pixels. This chip operates with three power supplies, analogue 2.8 V, digital 1.2 V, and IF 1.8 V, and has low power consumption. High sensitivity, low dark current, and no smear are achieved through the adoption of R, G, and B primary color pigment mosaic filters. This chip features an electronic shutter with variable charge-storage time.

In addition, this product is designed for use in cellular phone and tablet pc. When using this for another application, Sony does not guarantee the quality and reliability of product. Therefore, don't use this for applications other than cellular phone and tablet pc. Consult your Sony sales representative if you have any questions.

Features

- ◆ Back-illuminated CMOS image sensor Exmor R™
- ◆ 2-wire serial communication circuit on chip
- ◆ CSI2 serial data output (selection of 4lane/2lane)
- ◆ Timing generator, H and V driver circuits on chip
- ◆ CDS/PGA on chip
- ◆ 10-bit A/D converter on chip
- ◆ Automatic optical black (OB) clamp circuit on chip
- ◆ PLL on chip (rectangular wave)
- ◆ High sensitivity, low dark current, no smear
- ◆ Excellent anti-blooming characteristics
- ◆ Variable-speed shutter function (1 H units)
- ◆ R, G, B primary color pigment mosaic filters on chip
- ◆ Max. 30 frame/s in all-pixel scan mode
- ◆ Pixel rate: 280 [Mpixel/s] (All-pixels mode)
- ◆ 180 frame/s @720p with 2x2 analog (special) binning, 60 frame/s @1080p with V-crop
- ◆ Data rate: Max. 755 Mbps/lane(@4lane), 912Mbps/Lane(@2lane)

Exmor R

* Exmor R is a trademark of Sony Corporation. The Exmor R is a Sony's CMOS image sensor with significantly enhanced imaging characteristics including sensitivity and low noise by changing fundamental structure of Exmor™ pixel adopted column parallel A/D converter to back-illuminated type.

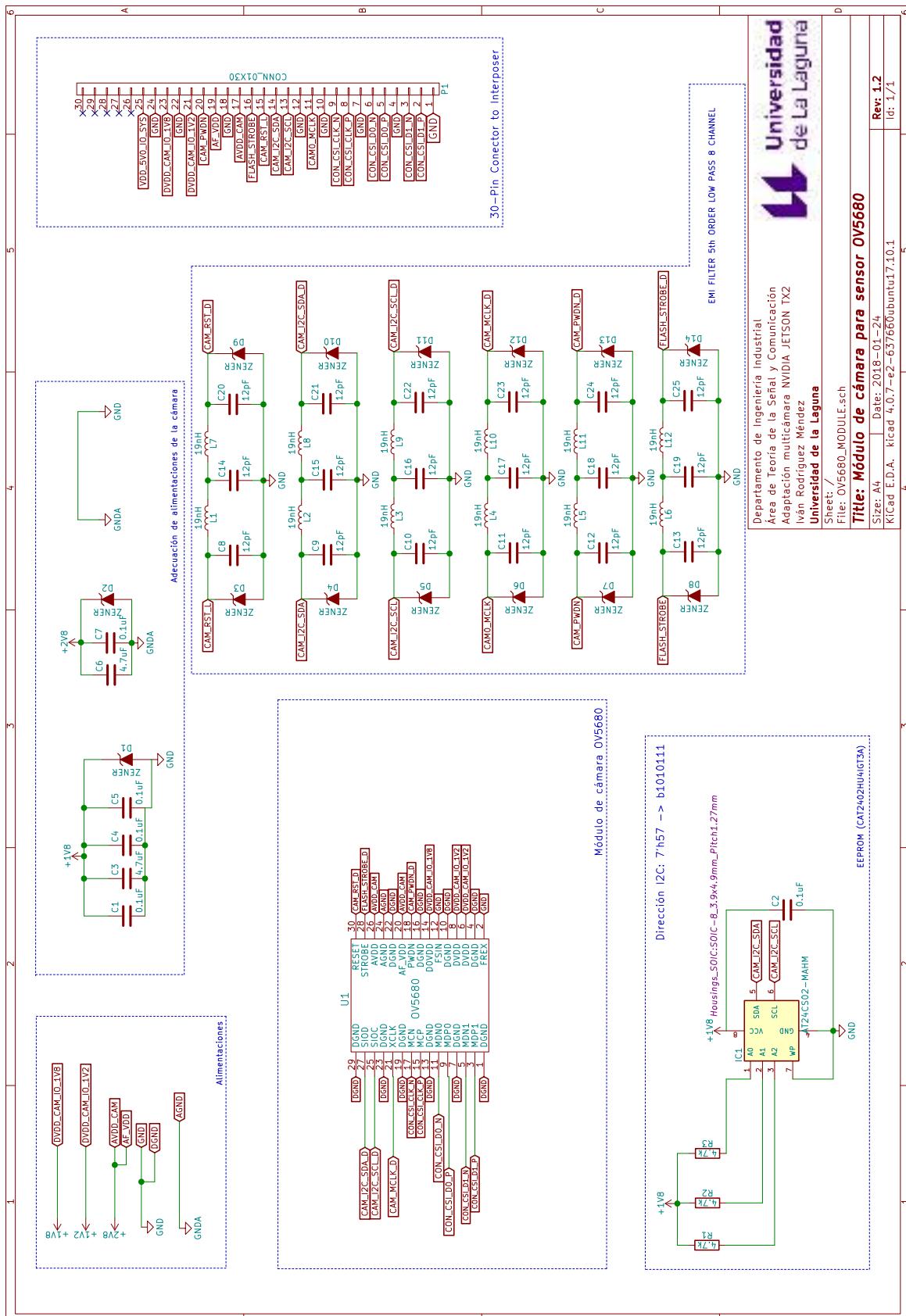
Sony reserves the right to change products and specifications without prior notice.
This information does not convey any license by any implication or otherwise under any patents or other right.
Application circuits shown, if any, are typical examples illustrating the operation of the devices. Sony cannot assume responsibility for any problems arising out of the use of these circuits.

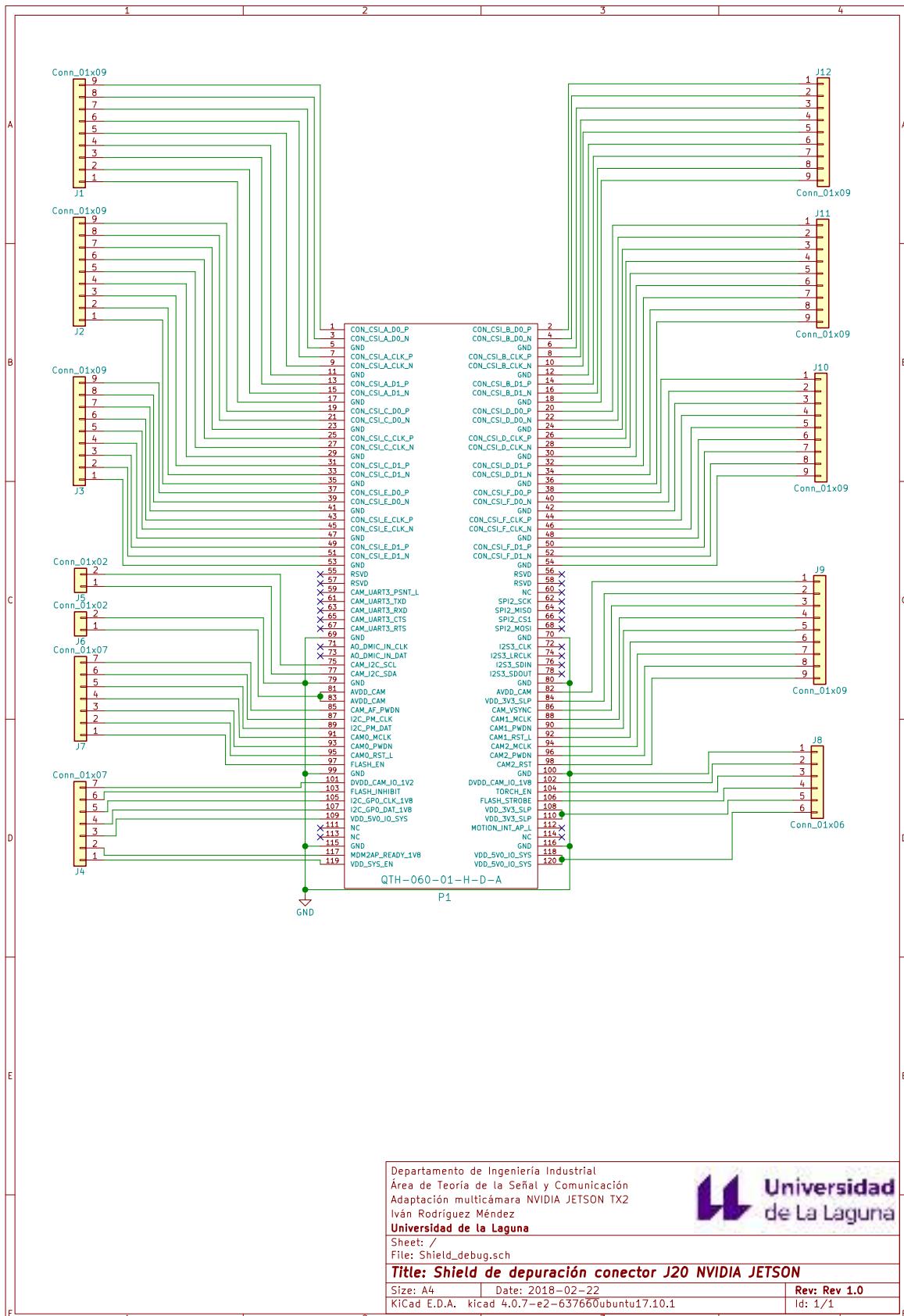
Anexo F

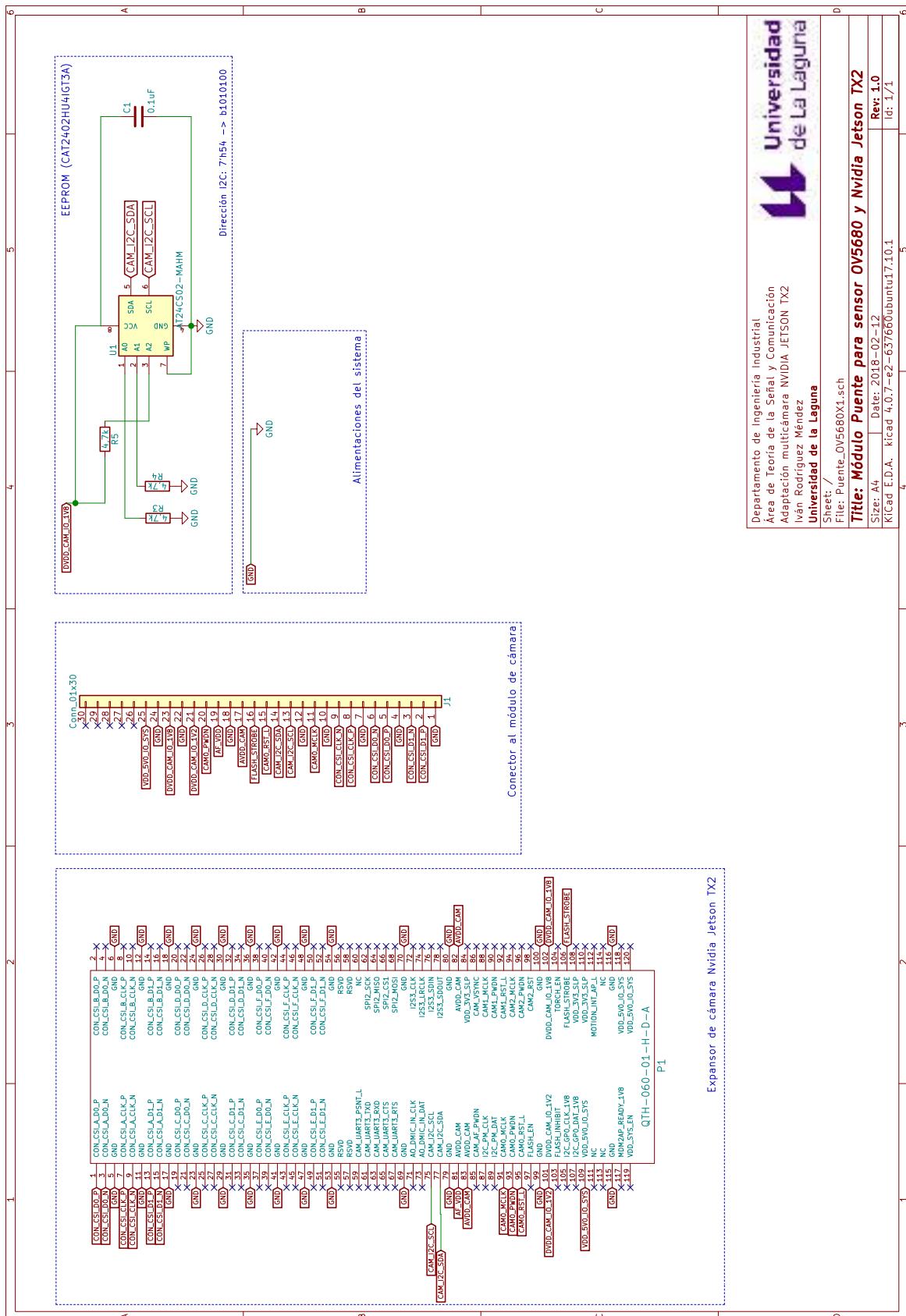
Hojas de esquema del diseño electrónico

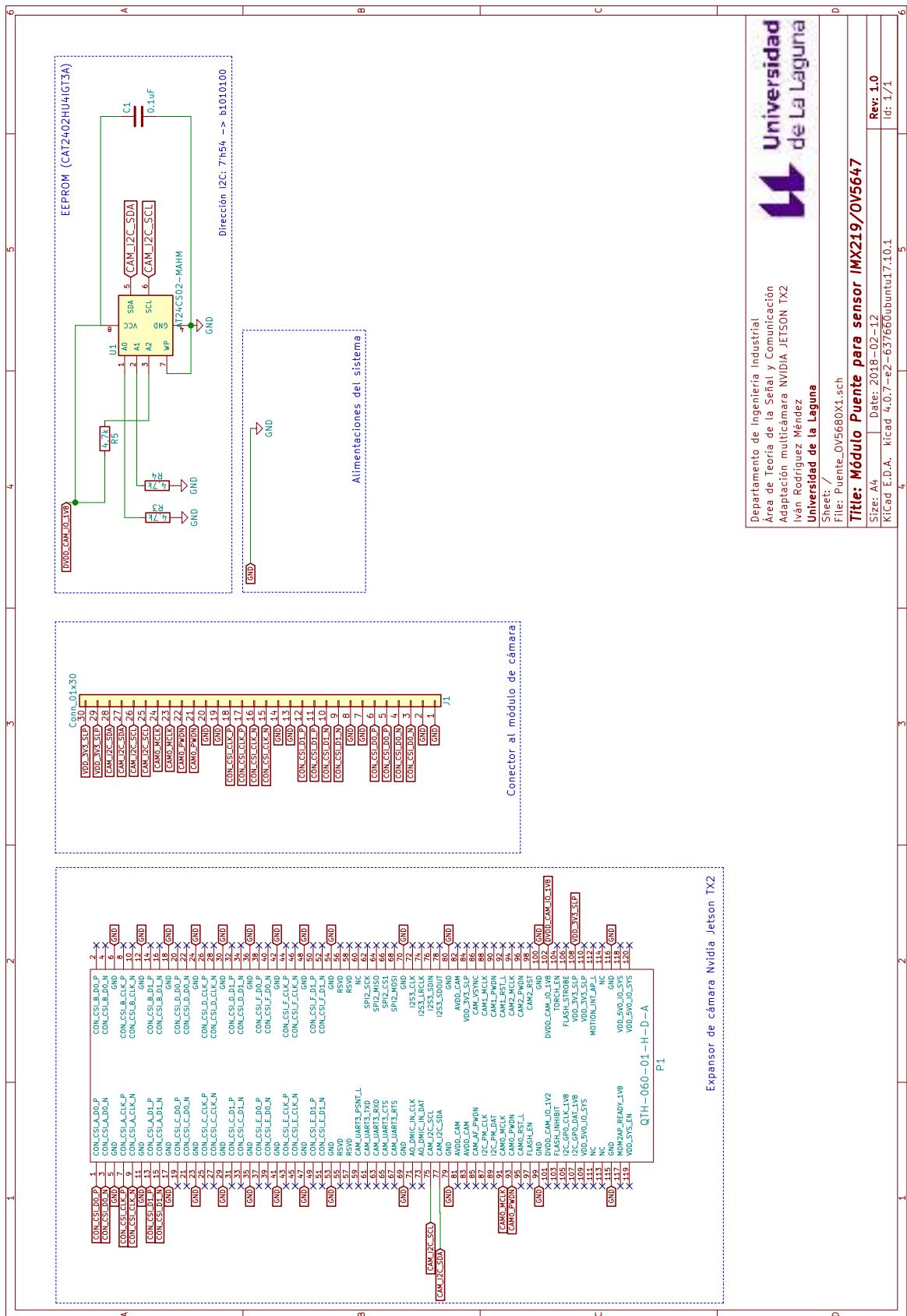
A continuación, se muestran los diferentes diseños electrónicos realizados durante el proyecto. Se mostrarán los diferentes diseños realizados siendo su orden el siguiente:

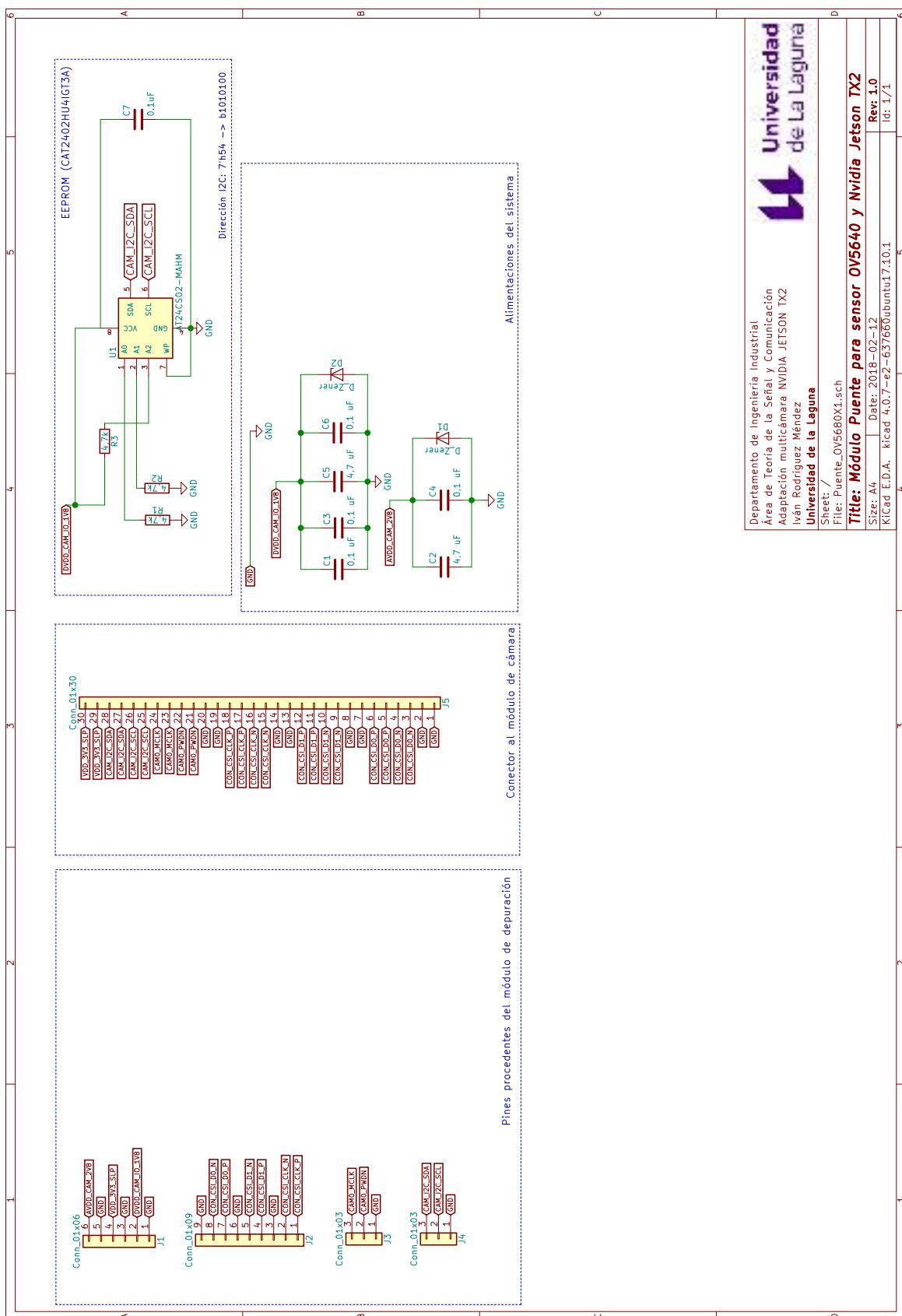
1. **Diseño para el sensor de cámara OV5680:** Diseño realizado para la adaptación del sensor de cámara OV5680 en la NVIDIA JETSON.
2. **Módulo de depuración de líneas:** Módulo de depuración de líneas del conector J22 de la NVIDIA JETSON.
3. **Módulo puente sensor OV5680:** Módulo puente para la conexión del módulo de cámara a la NVIDIA JETSON.
4. **Módulo puente para el sensor IMX219/OV5647:** Módulo puente para los sensores IMX219 y OV5647 de Raspberry Pi.
5. **Módulo puente con conexión DUPONT:** Módulo puente con conexión a pines DUPONT para los sensores IMX219 y OV5647.











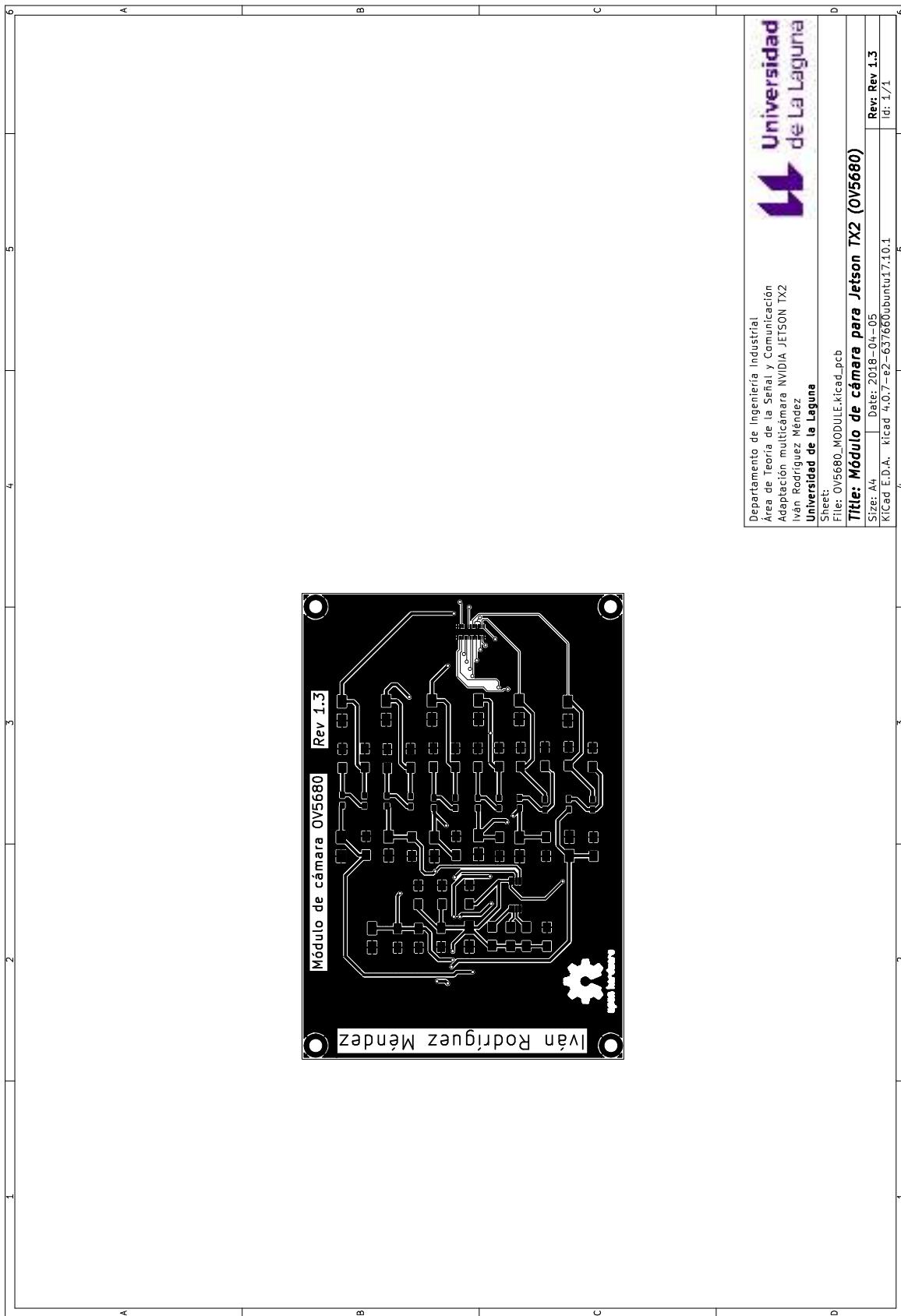
Anexo G

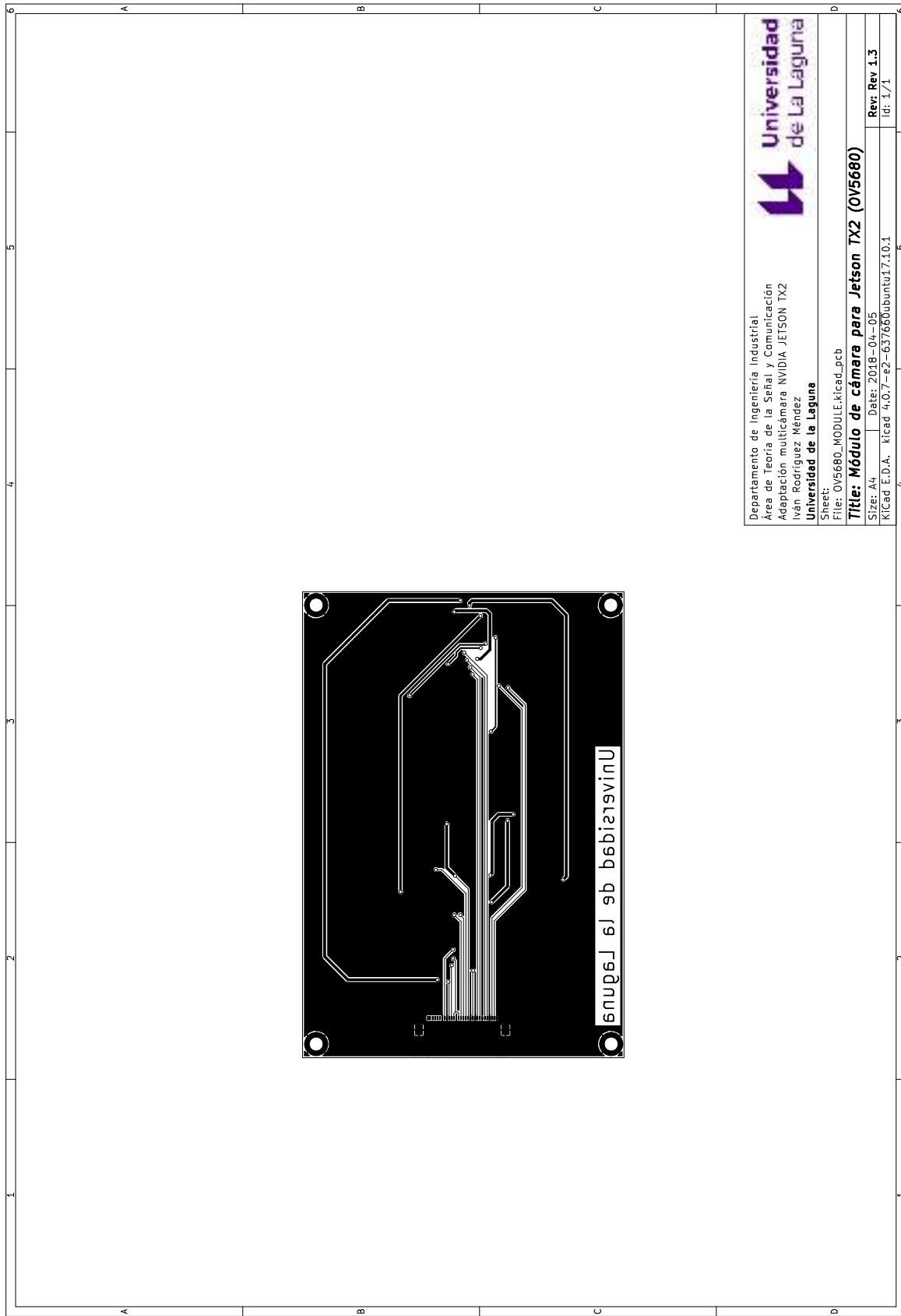
Planos de diseño de placas de circuito impreso

A continuación, se muestran los fotolitos y planos requeridos para la fabricación de las placas de circuito impreso. Se incluyen las láminas conductoras de las capas *top* y *bottom*, además de la colocación de los elementos, la máscara de soldadura y la localización de los taladros. El orden de los planos presentados en este anexo es el siguiente:

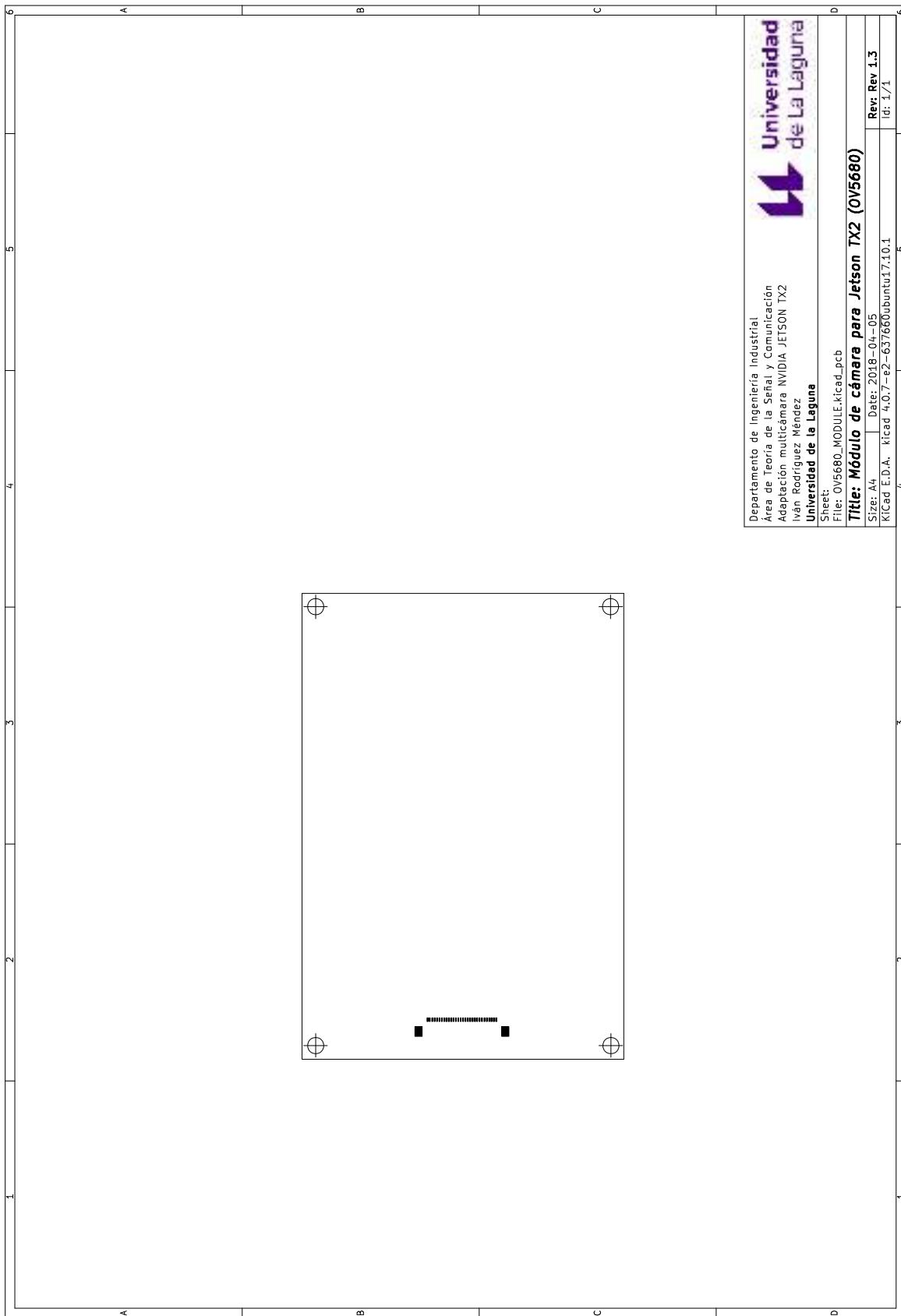
1. **Diseño para el sensor de cámara OV5680:** PCB realizada para la adaptación del sensor de cámara OV5680 en la NVIDIA JETSON.
2. **Módulo puente sensor OV5680:** Módulo puente para la conexión del módulo de cámara a la NVIDIA JETSON.
3. **Módulo puente para el sensor IMX219/OV5647:** Módulo puente para los sensores IMX219 y OV5647 de Raspberry Pi.
4. **Módulo puente con conexión DUPONT:** Módulo puente con conexión a pines DUPONT para los sensores IMX219 y OV5647.
5. **Módulo de depuración de líneas:** Módulo de depuración de líneas del conector J20 de la NVIDIA JETSON.
6. **Módulo de test de CNC:** Módulo para el test de fabricación de PCBs en CNCs.

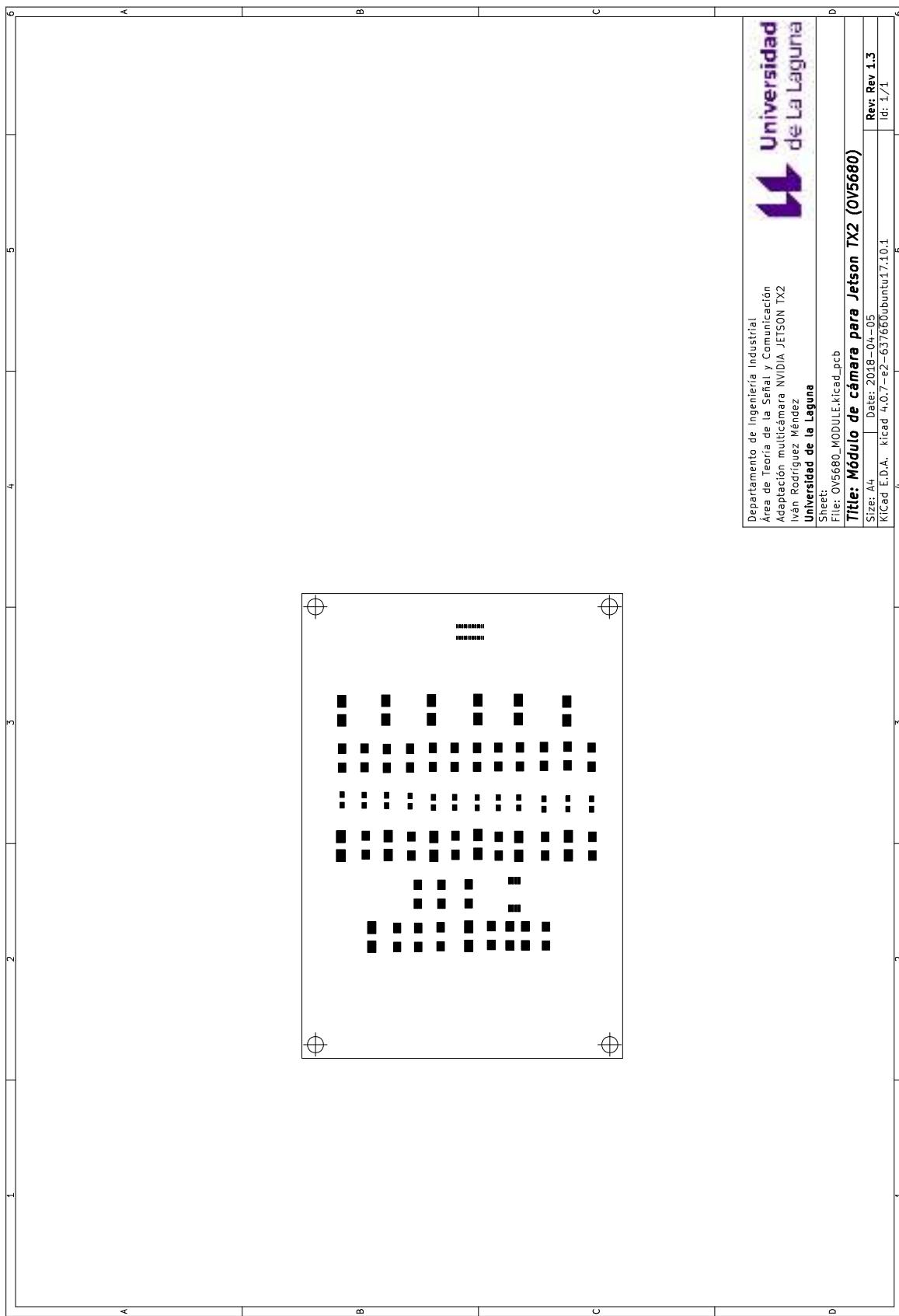
198 ANEXO G. PLANOS DE DISEÑO DE PLACAS DE CIRCUITO IMPRESO



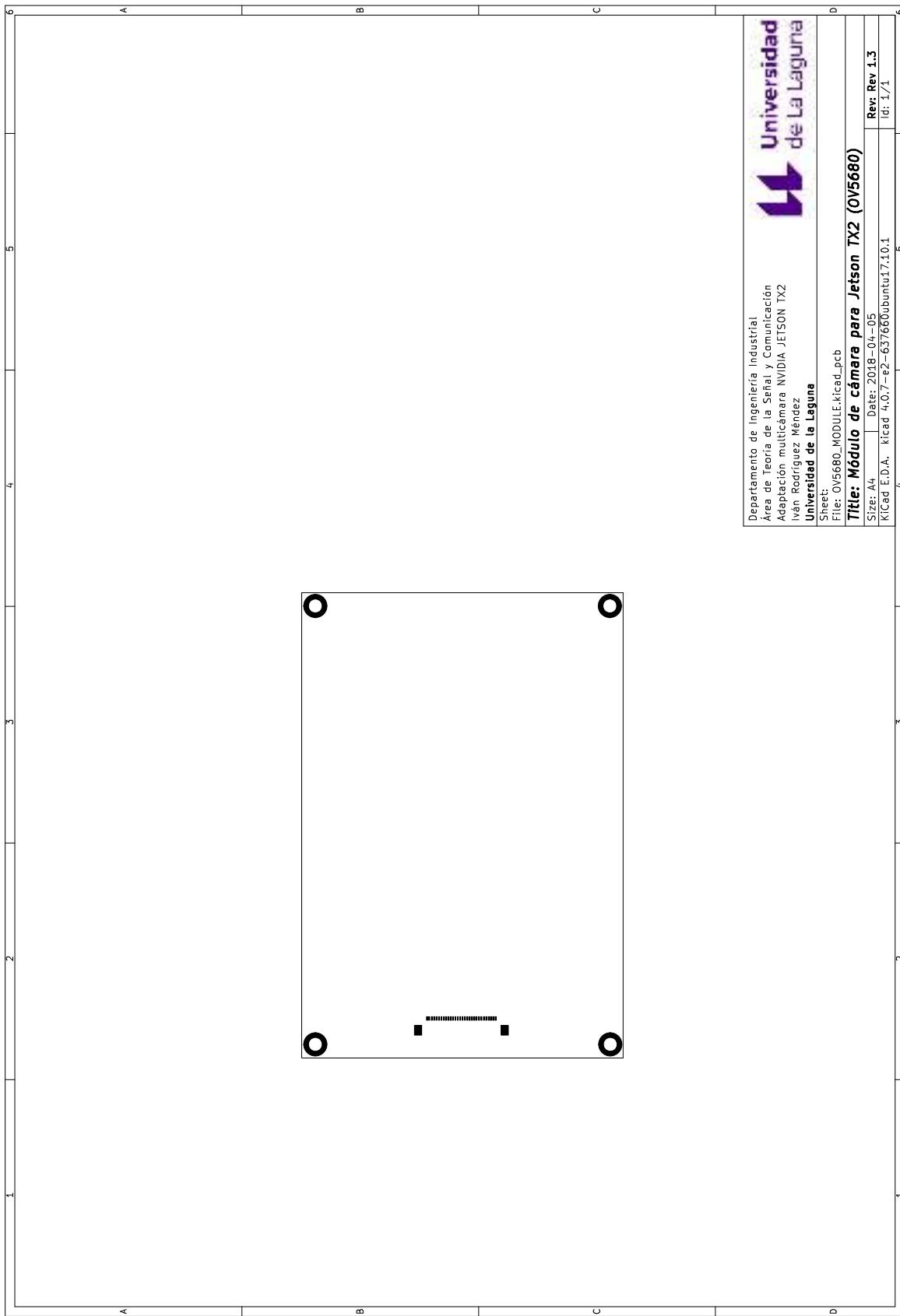


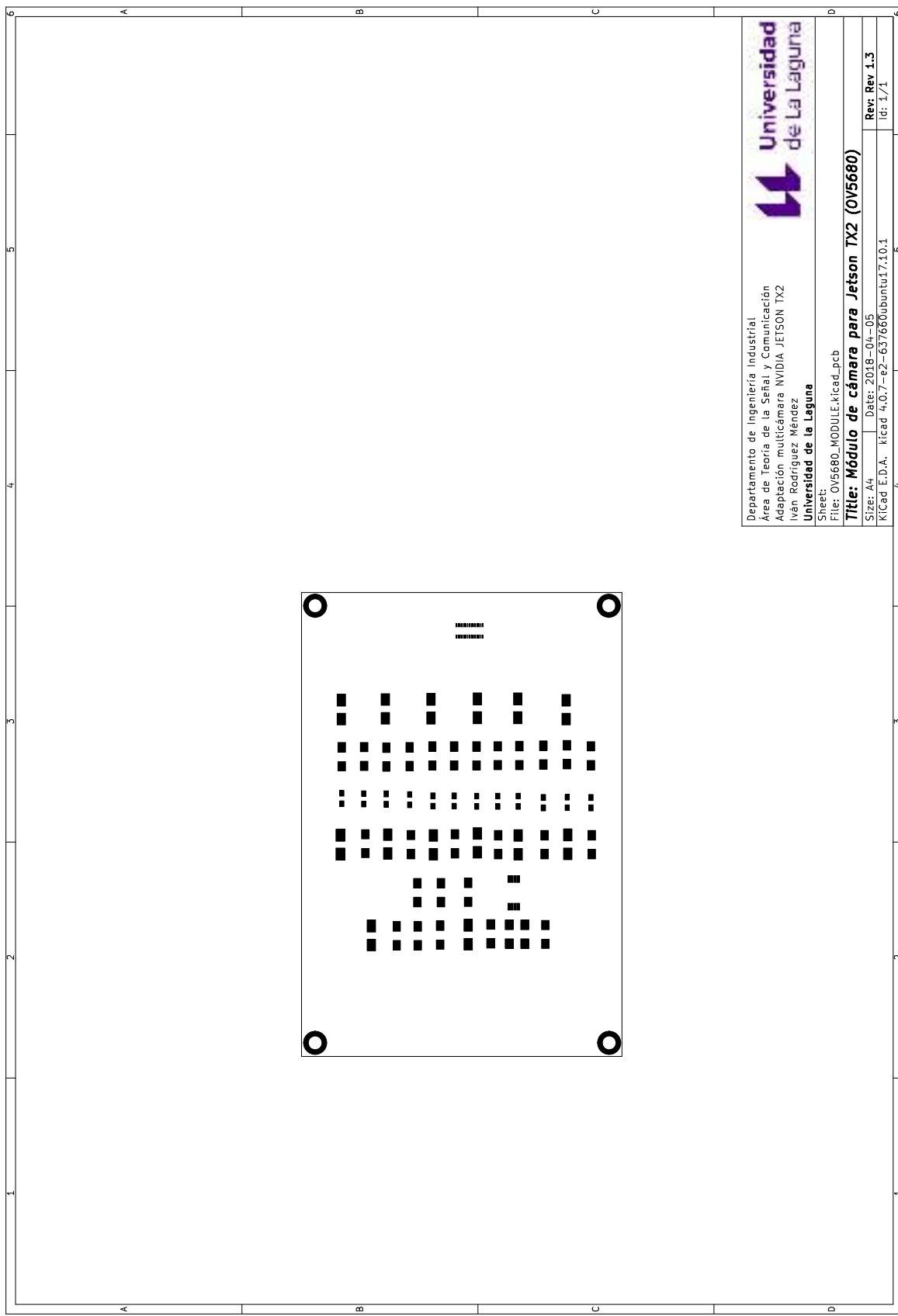
200ANEXO G. PLANOS DE DISEÑO DE PLACAS DE CIRCUITO IMPRESO



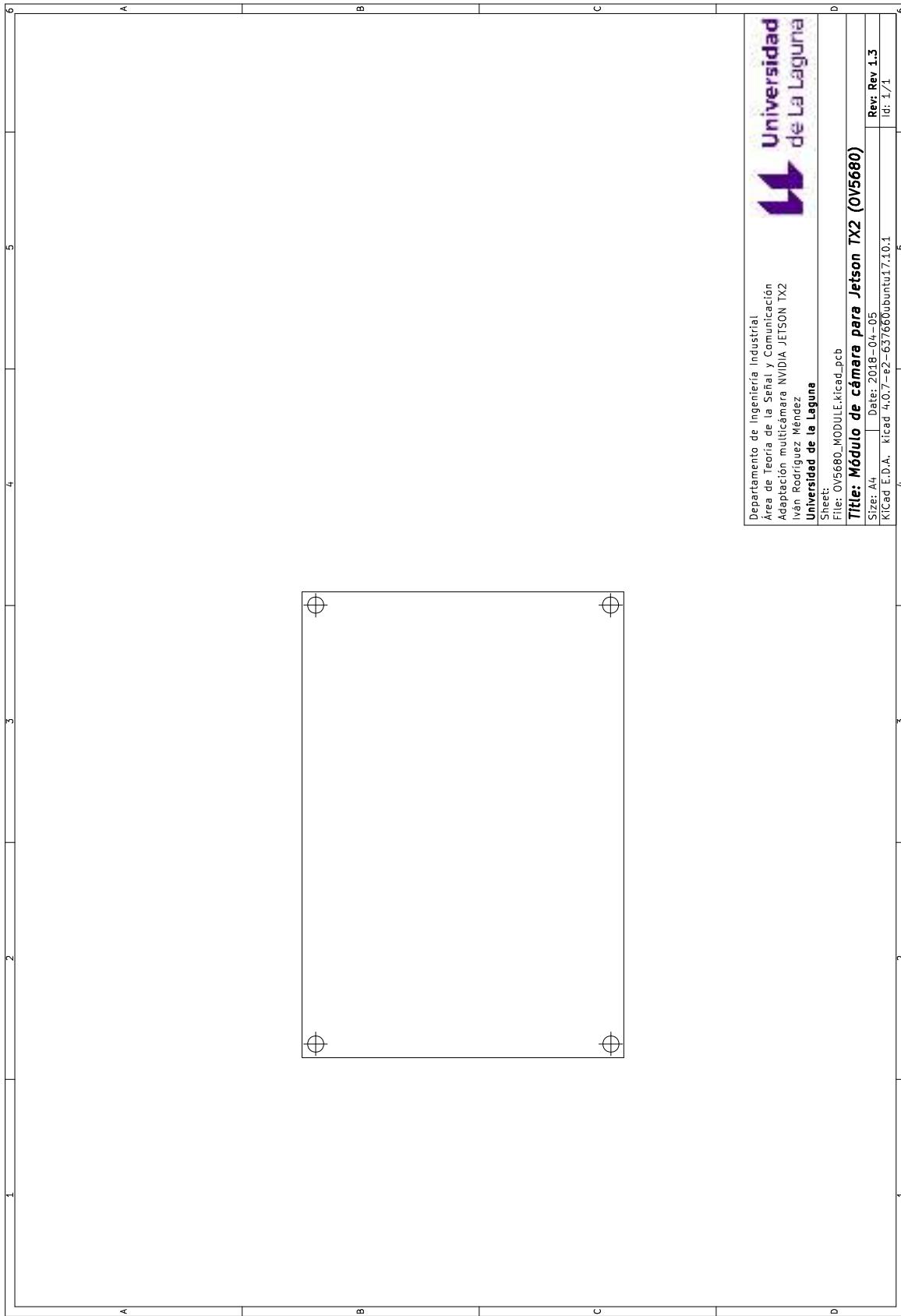


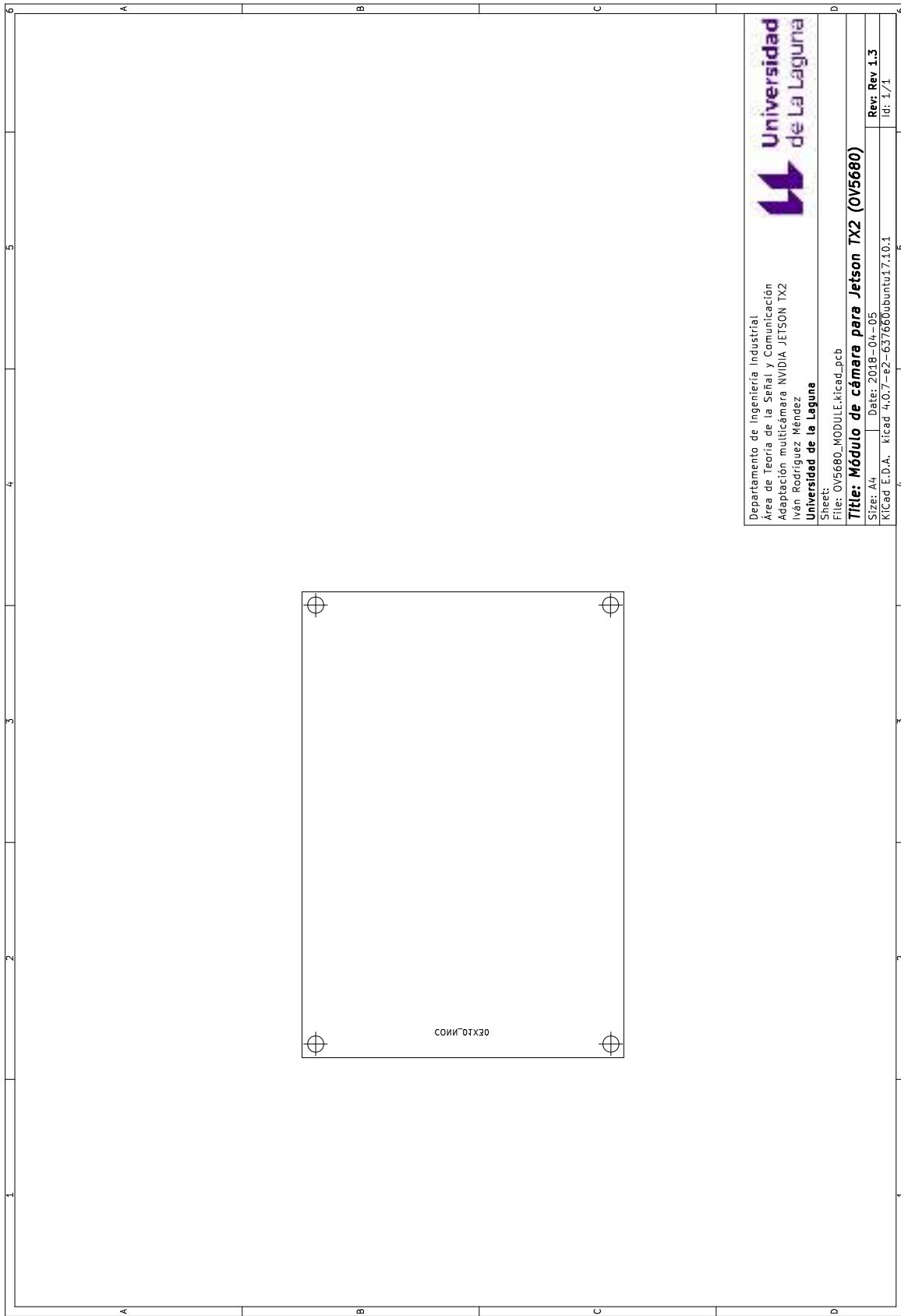
202 ANEXO G. PLANOS DE DISEÑO DE PLACAS DE CIRCUITO IMPRESO



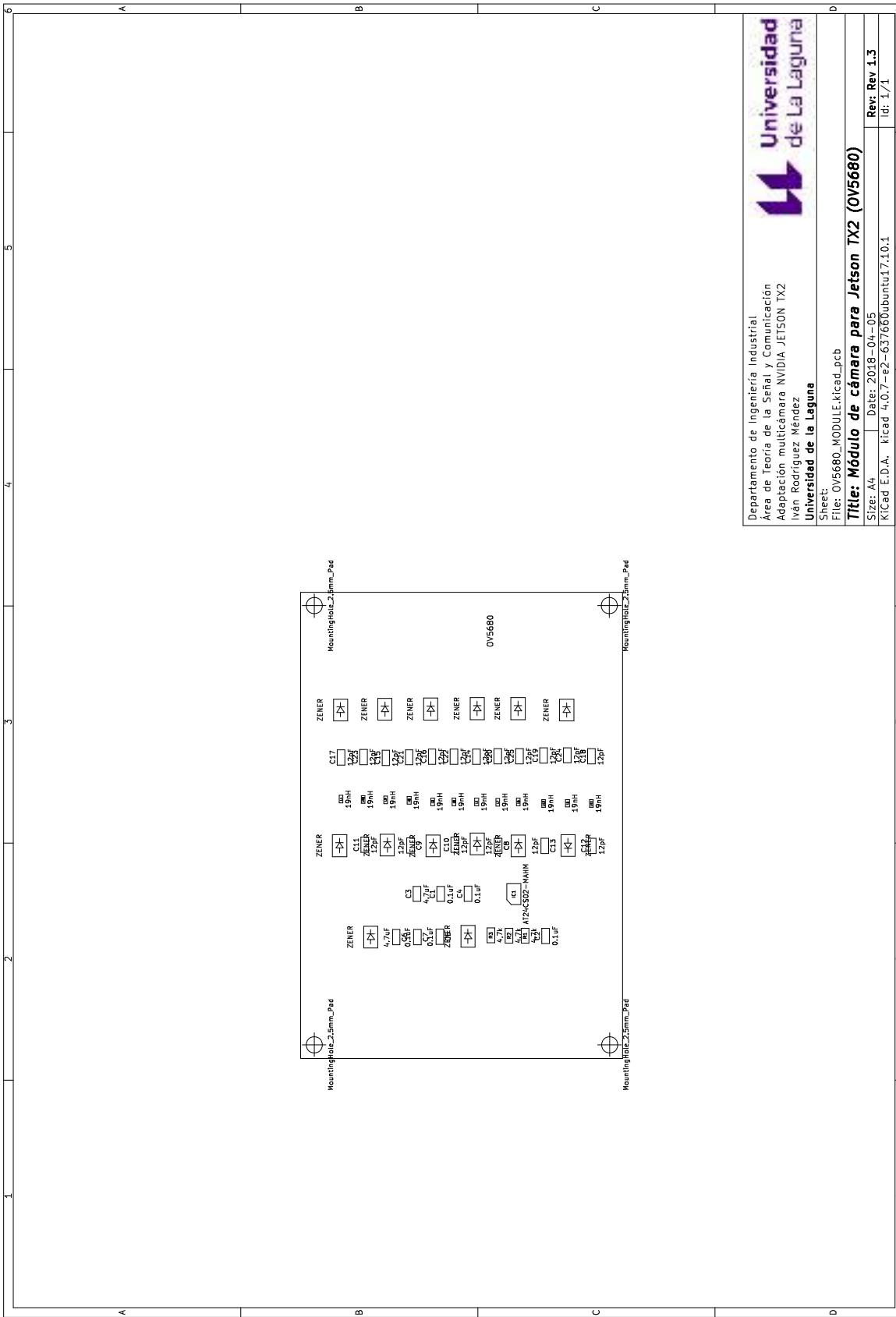


204 ANEXO G. PLANOS DE DISEÑO DE PLACAS DE CIRCUITO IMPRESO





206ANEXO G. PLANOS DE DISEÑO DE PLACAS DE CIRCUITO IMPRESO



Departamento de Ingeniería Industrial
Área de Teoría de la Señal y Comunicación
Adaptación multicámaras NVIDIA JETSON TX2
Ivan Rodriguez Vélez
Universidad de La Laguna

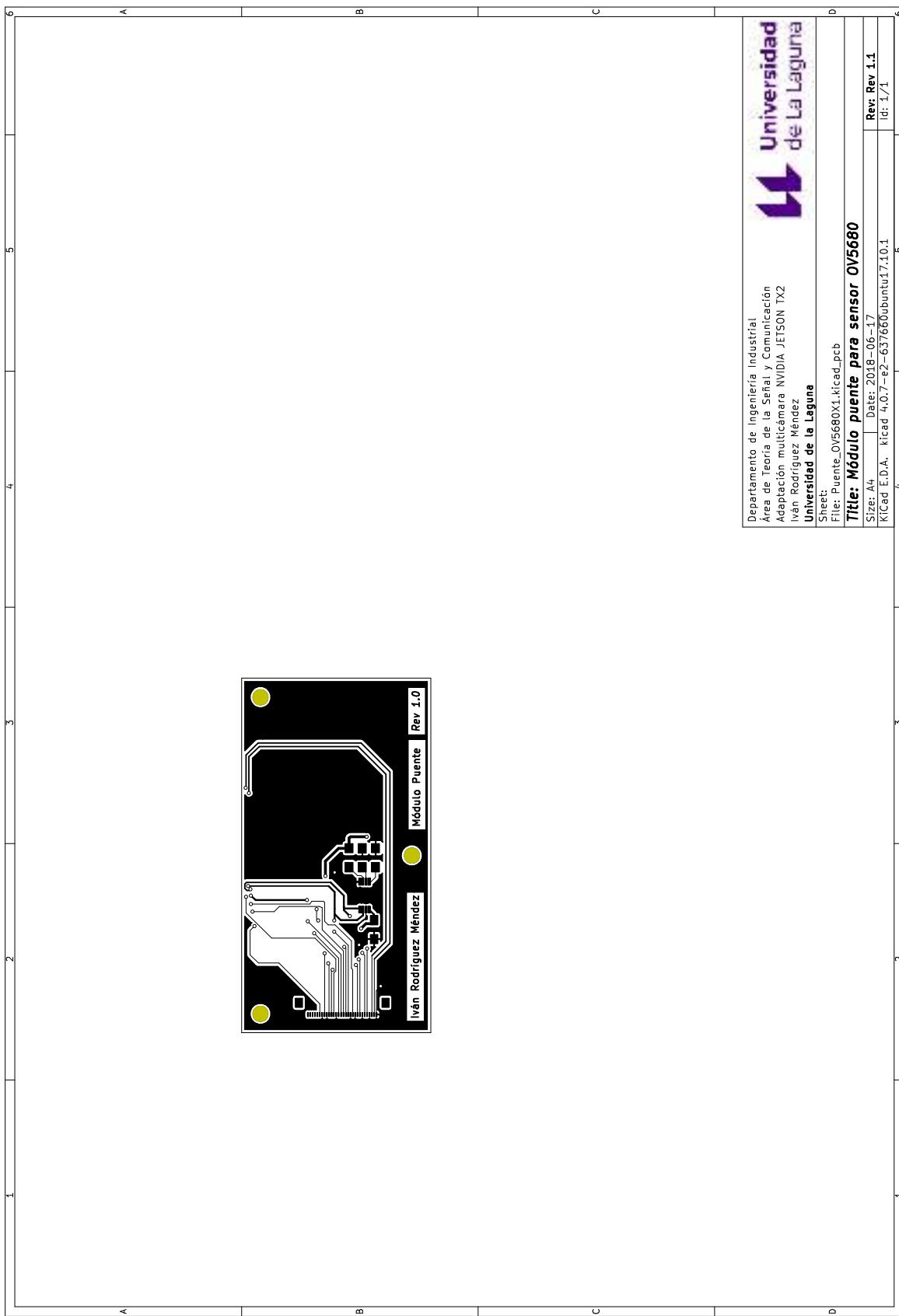
Sheet:

File: OV5680_MODULE.kicad_pcb

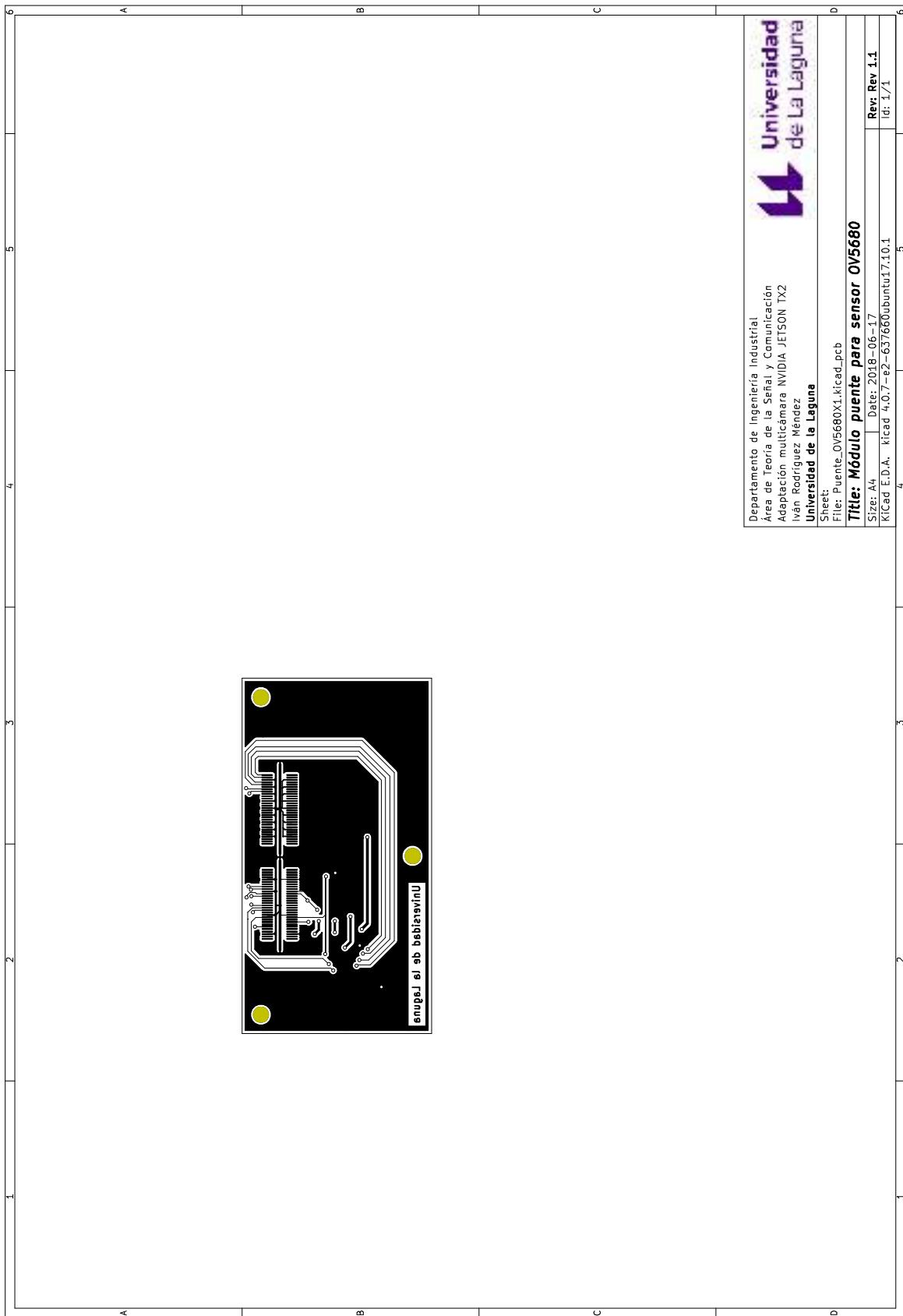
Title: Módulo de cámara para Jetson TX2 (OV5680)

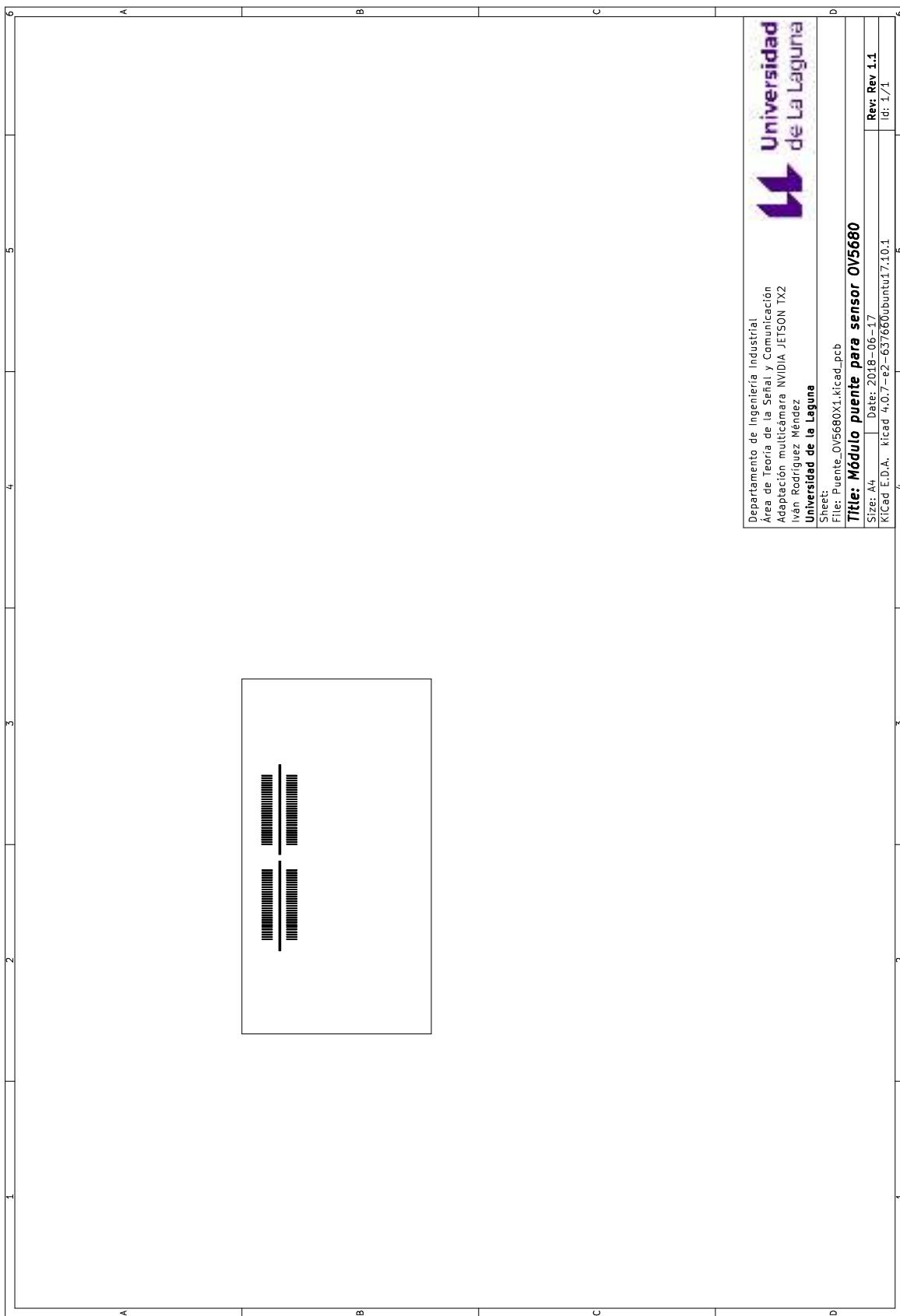
Size: A4 Date: 2018-04-05 Rev: Rev 1.3

Id: 1/1

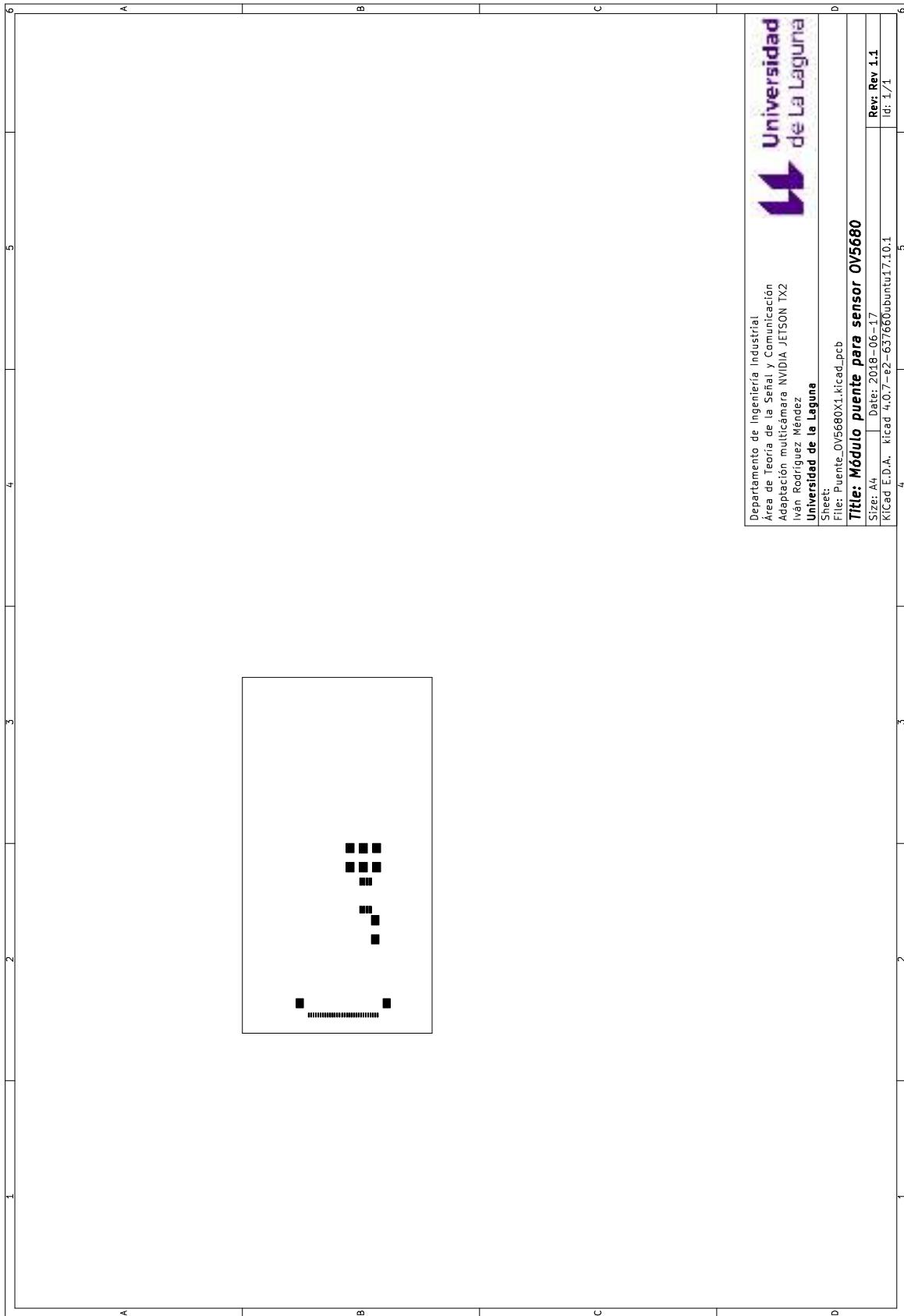


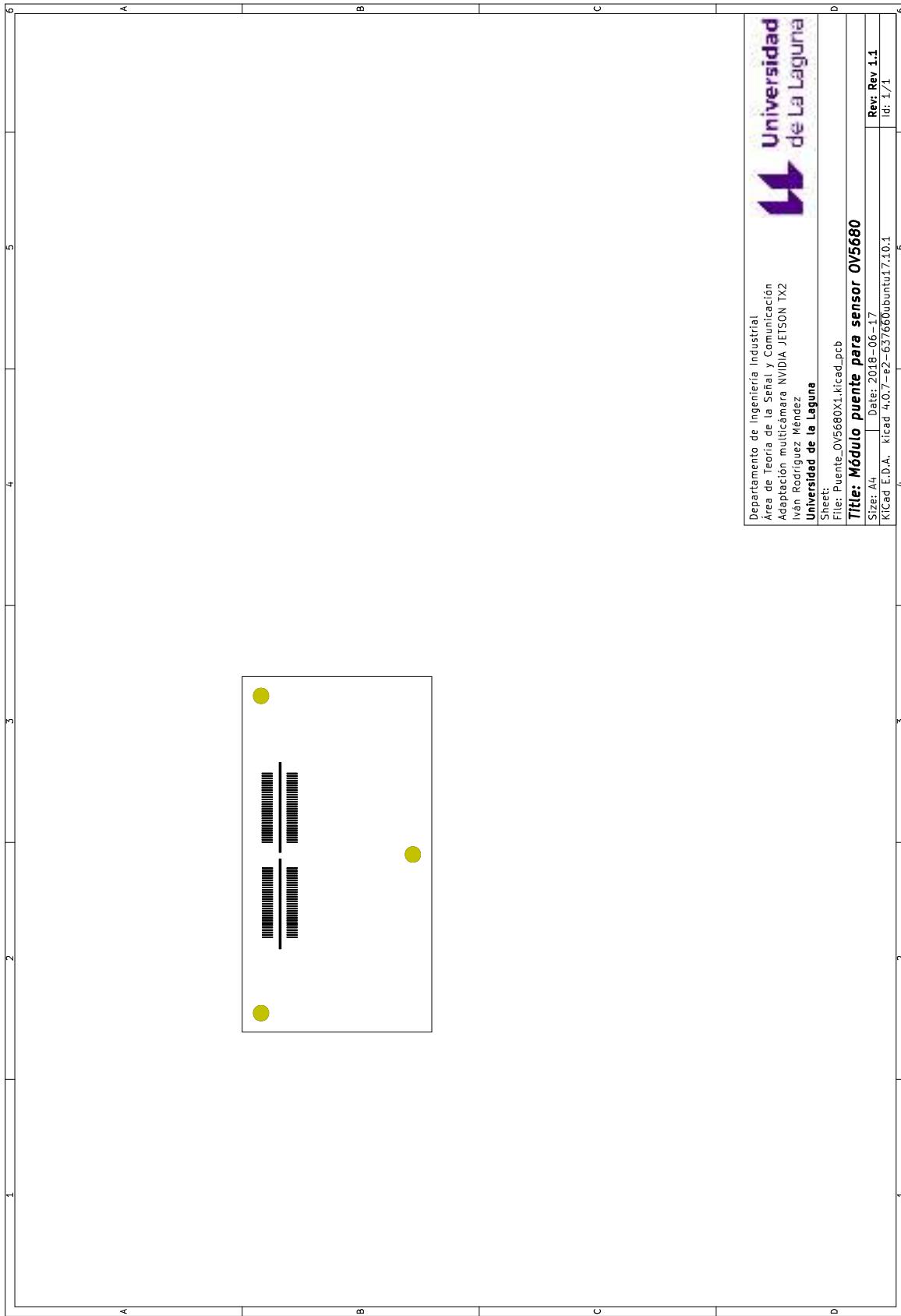
208 ANEXO G. PLANOS DE DISEÑO DE PLACAS DE CIRCUITO IMPRESO



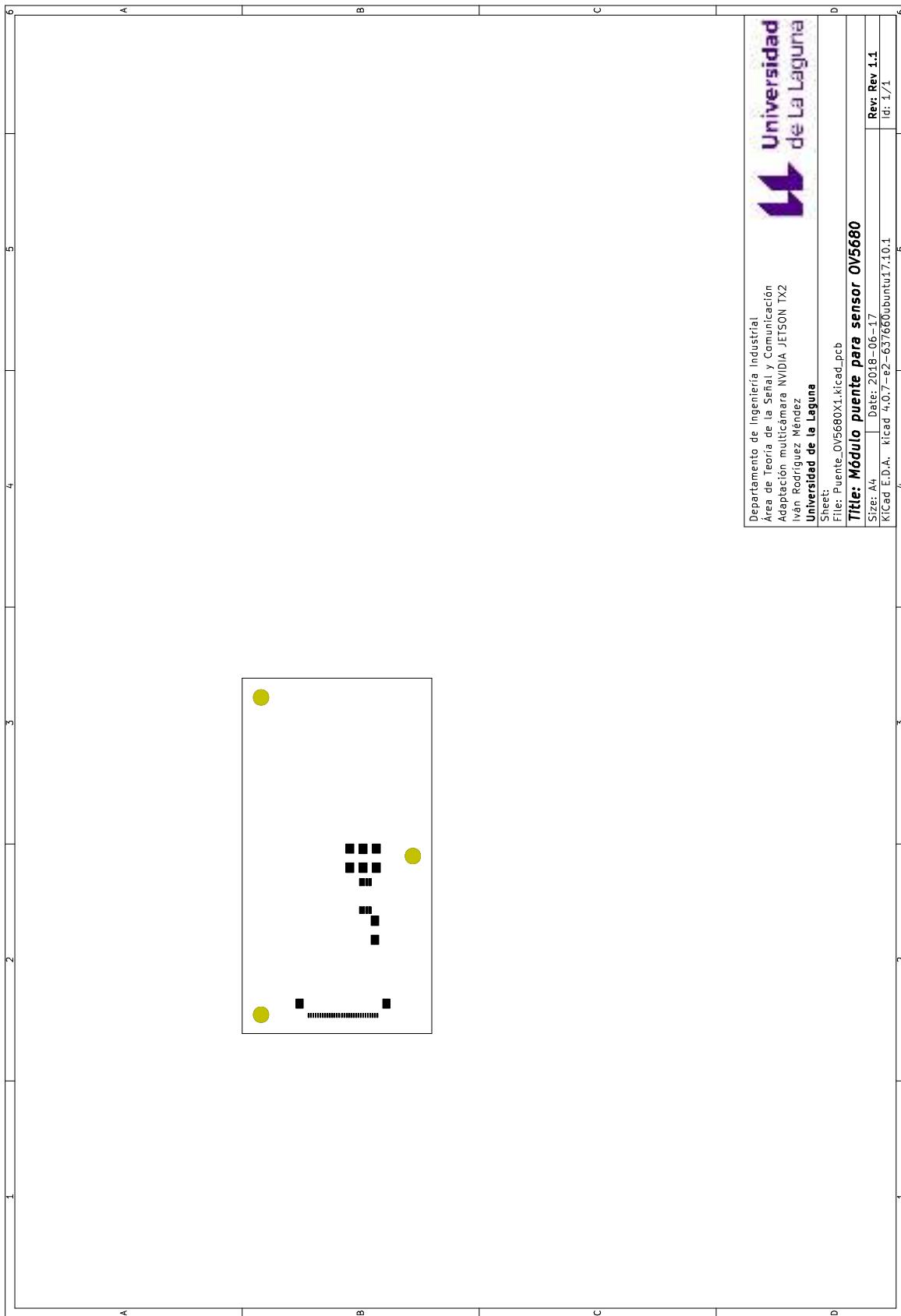


210ANEXO G. PLANOS DE DISEÑO DE PLACAS DE CIRCUITO IMPRESO





212ANEXO G. PLANOS DE DISEÑO DE PLACAS DE CIRCUITO IMPRESO



| | | | | | | | |
|---|---|--|---|--|---|--|---|
| | A | | B | | C | | D |
| 6 | | | | | | | |
| 5 | | | | | | | |
| 4 | | | | | | | |
| 3 | | | | | | | |
| 2 | | | | | | | |
| 1 | | | | | | | |
| | | | | | | | |

Universidad de La Laguna

Departamento de Ingeniería Industrial
Área de Teoría de la Señal y Comunicación
Adaptación multicámaras NVIDIA JETSON TX2
Ivan Rodriguez Vélez
Universidad de la Laguna

Sheet:
File: Puente_0V5680X4.Vicad.Lpcb

Title: Módulo puente para sensor 0V5680

| | | |
|---|------------------|--------------|
| Size: A4 | Date: 2018-06-17 | Rev: Rev 1.1 |
| KICad EDA, kitad 4.0.7-e2-637660ubuntu17.10.1 | | Id: 1/1 |
| 4 | 3 | 2 |
| 1 | | |

214ANEXO G. PLANOS DE DISEÑO DE PLACAS DE CIRCUITO IMPRESO

| | | | |
|---|---|---|---|
| A | B | C | D |
| 6 | | | |
| 5 | | | |
| 4 | | | |
| 3 | | | |
| 2 | | | |
| 1 | | | |
| 0 | | | |
| <p>A-0-H-10-00-000-H10</p> | | | |
|  | | | |
| A | B | C | D |

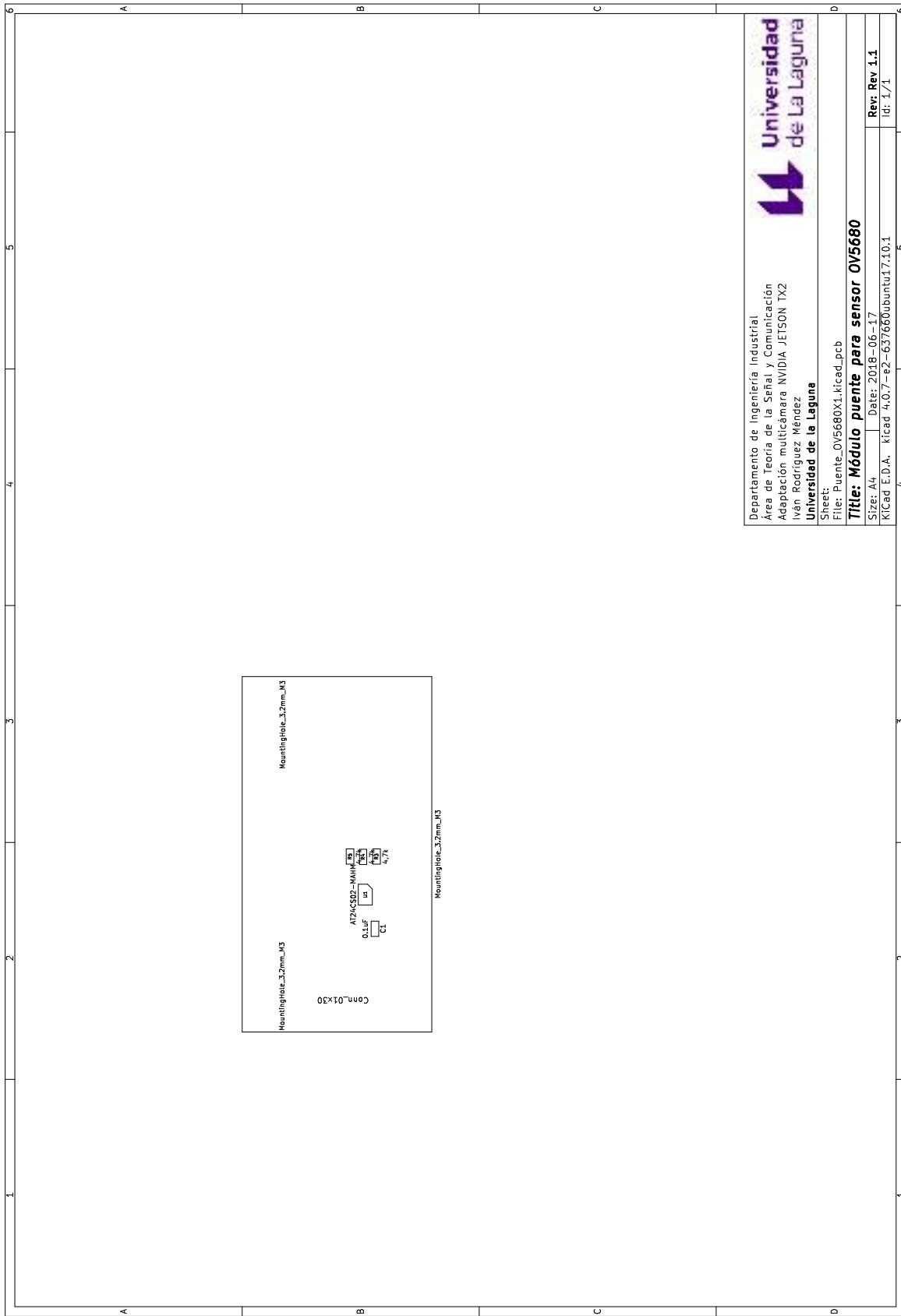
Universidad de La Laguna

Departamento de Ingeniería Industrial
Área de Teoría de la Señal y Comunicación
Adaptación multicámaras NVIDIA JETSON TX2
Ivan Rodriguez Vélez
Universidad de la Laguna

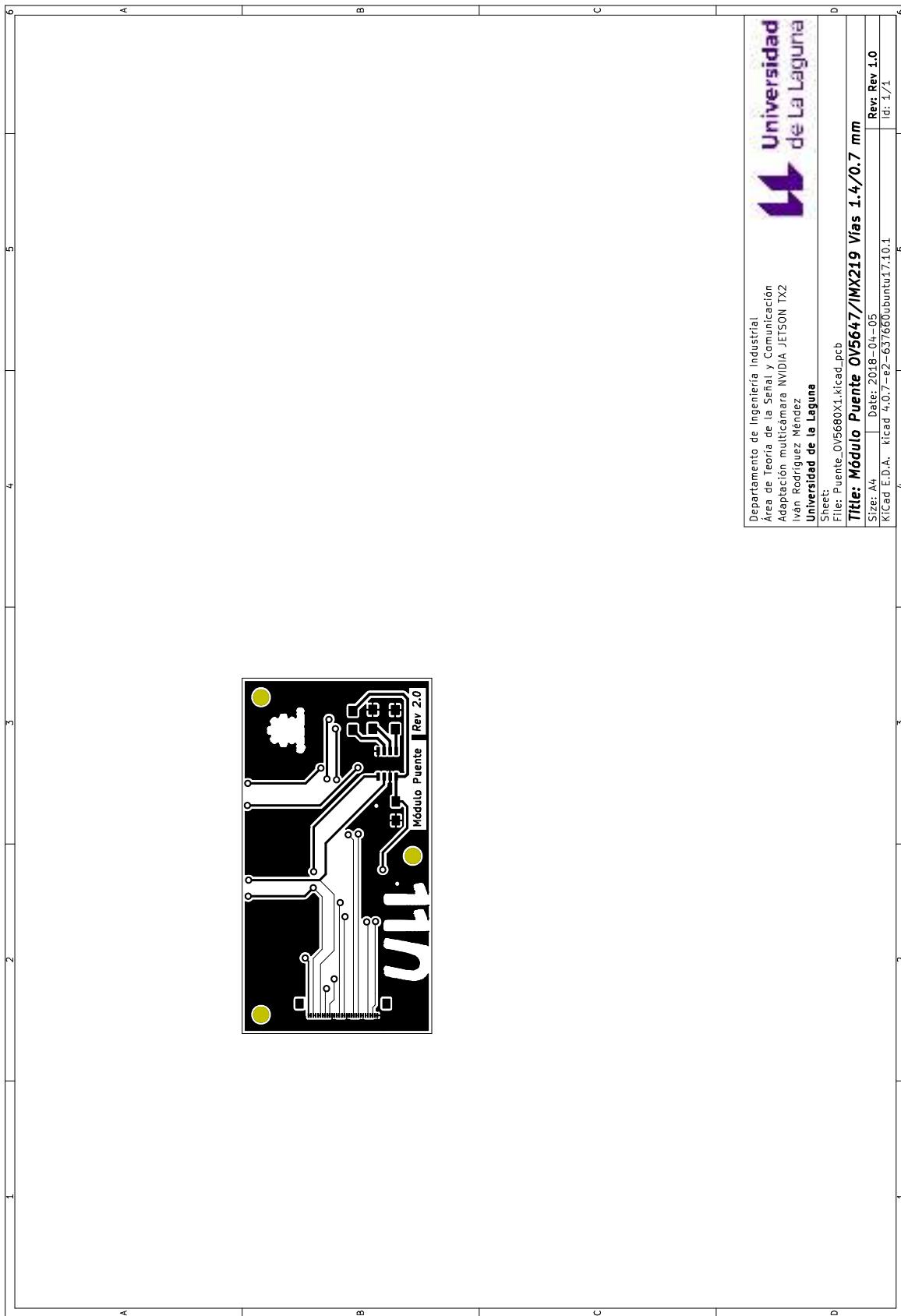
Sheet:
File: Puente_0V5680X4.Vicad.pcb

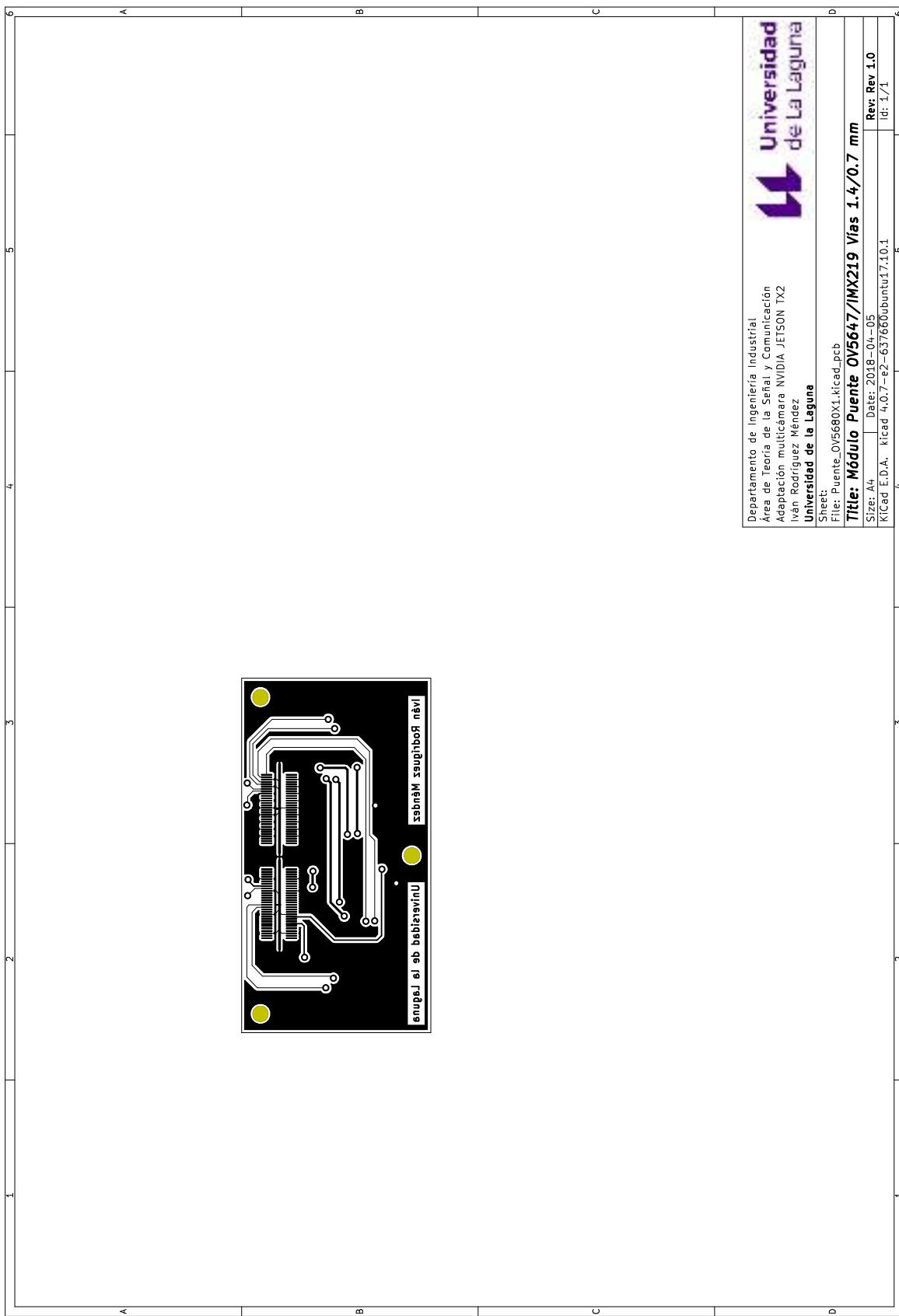
Title: Módulo puente para sensor 0V5680

| | | |
|---|------------------|--------------|
| Size: A4 | Date: 2018-06-17 | Rev: Rev 1.1 |
| KICad EDA, kitad 4.0.7-e2-637660ubuntu17.10.1 | | Id: 1/1 |
| 4 | 3 | 2 |
| 3 | 1 | |

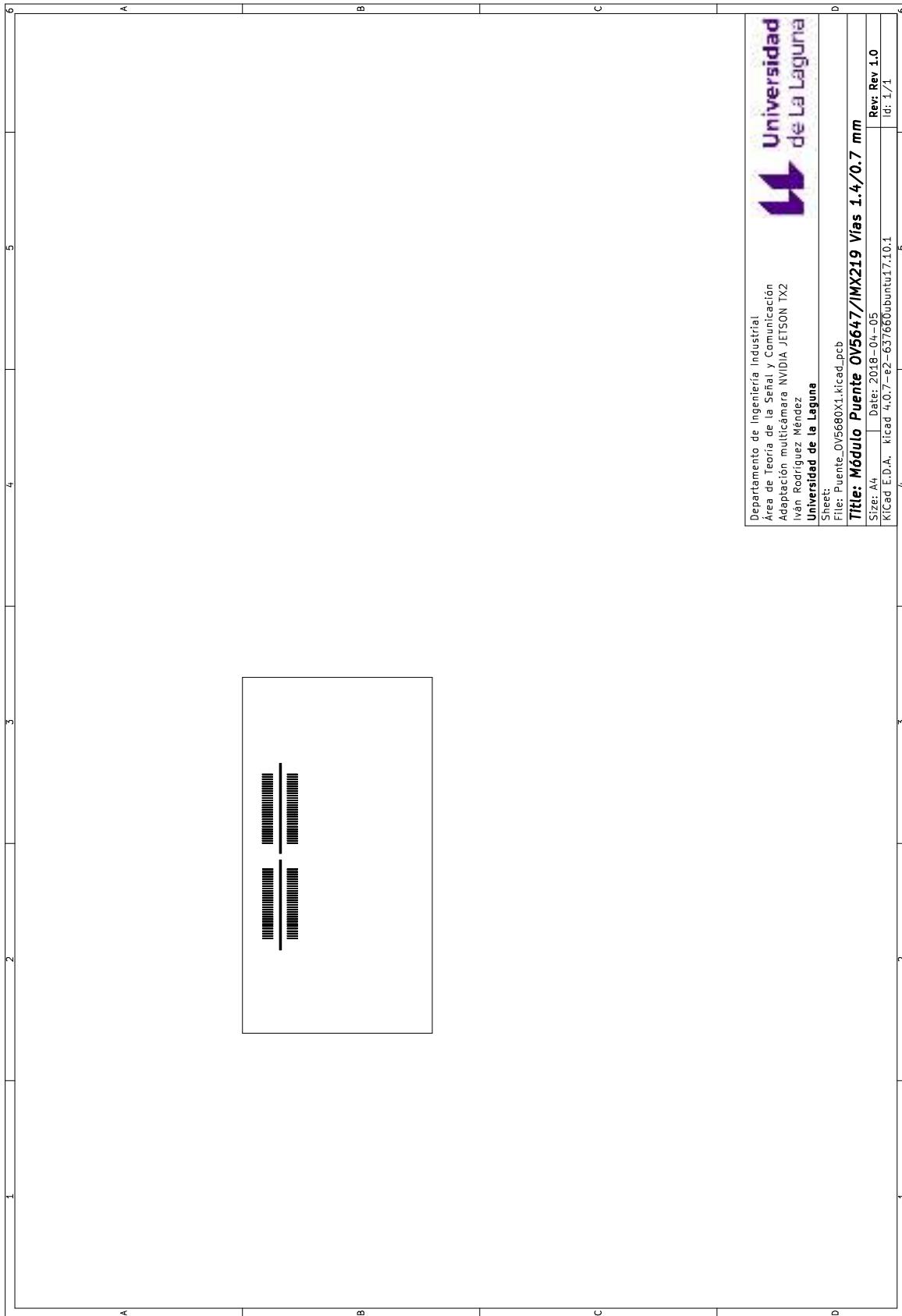


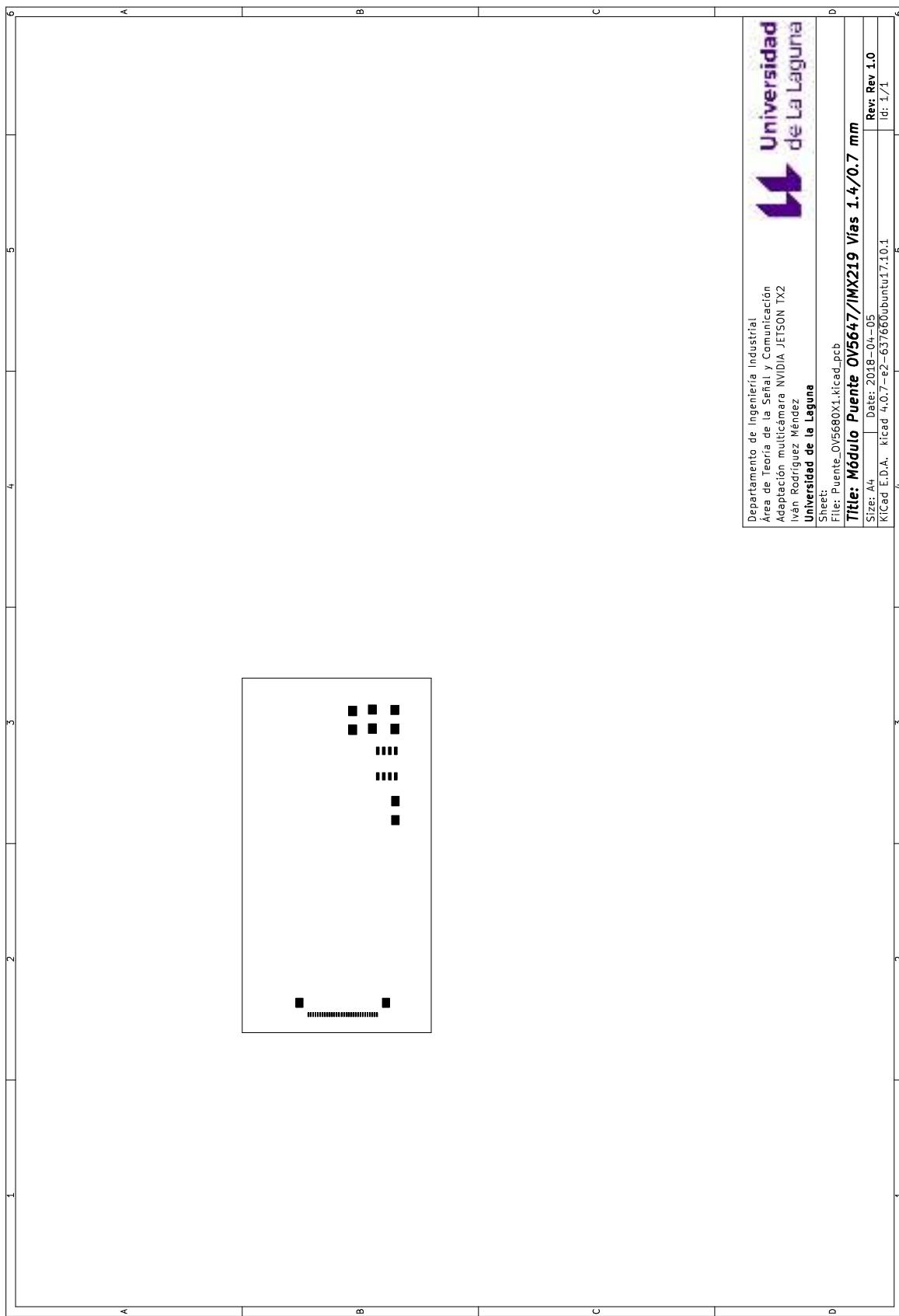
216ANEXO G. PLANOS DE DISEÑO DE PLACAS DE CIRCUITO IMPRESO



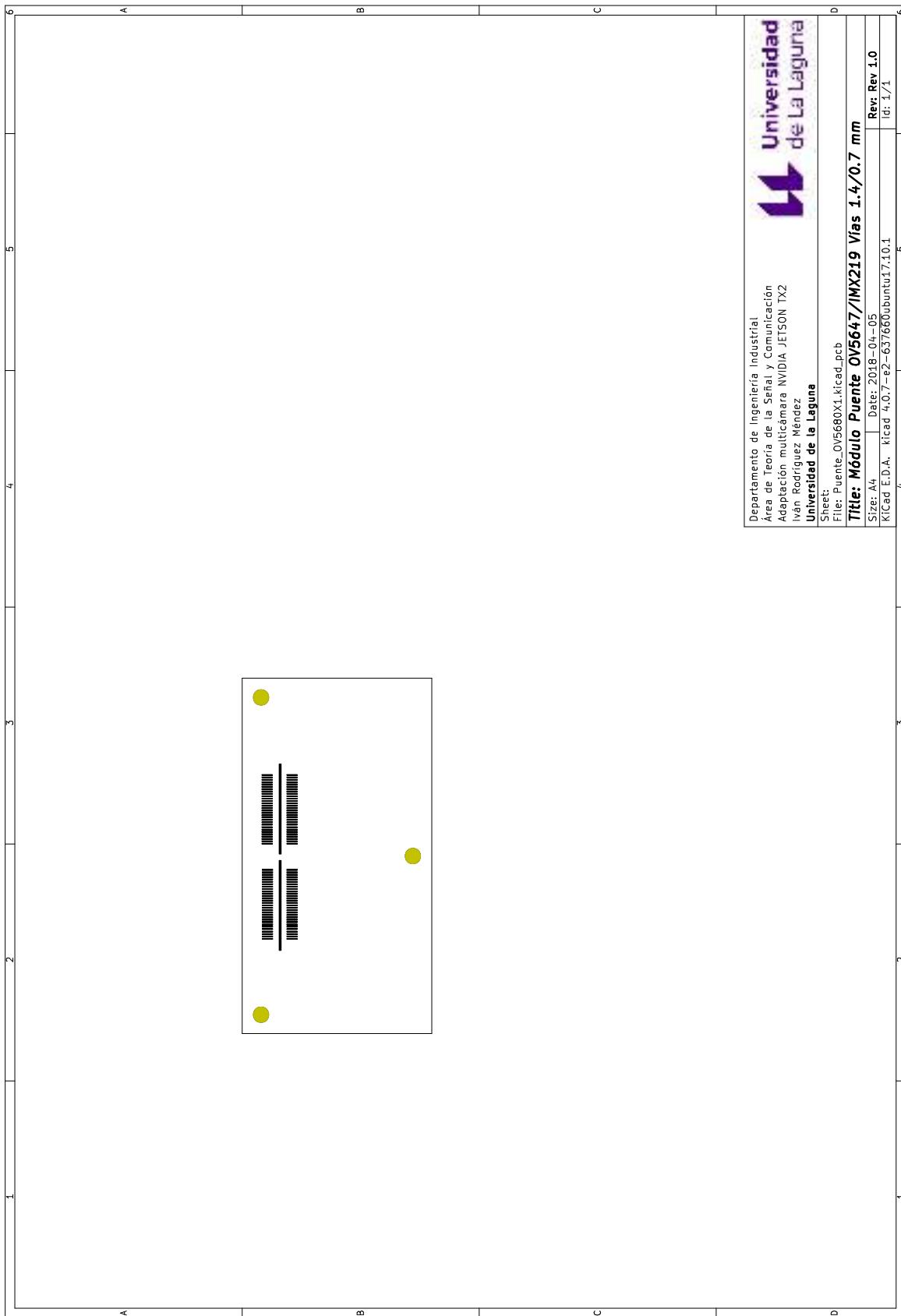


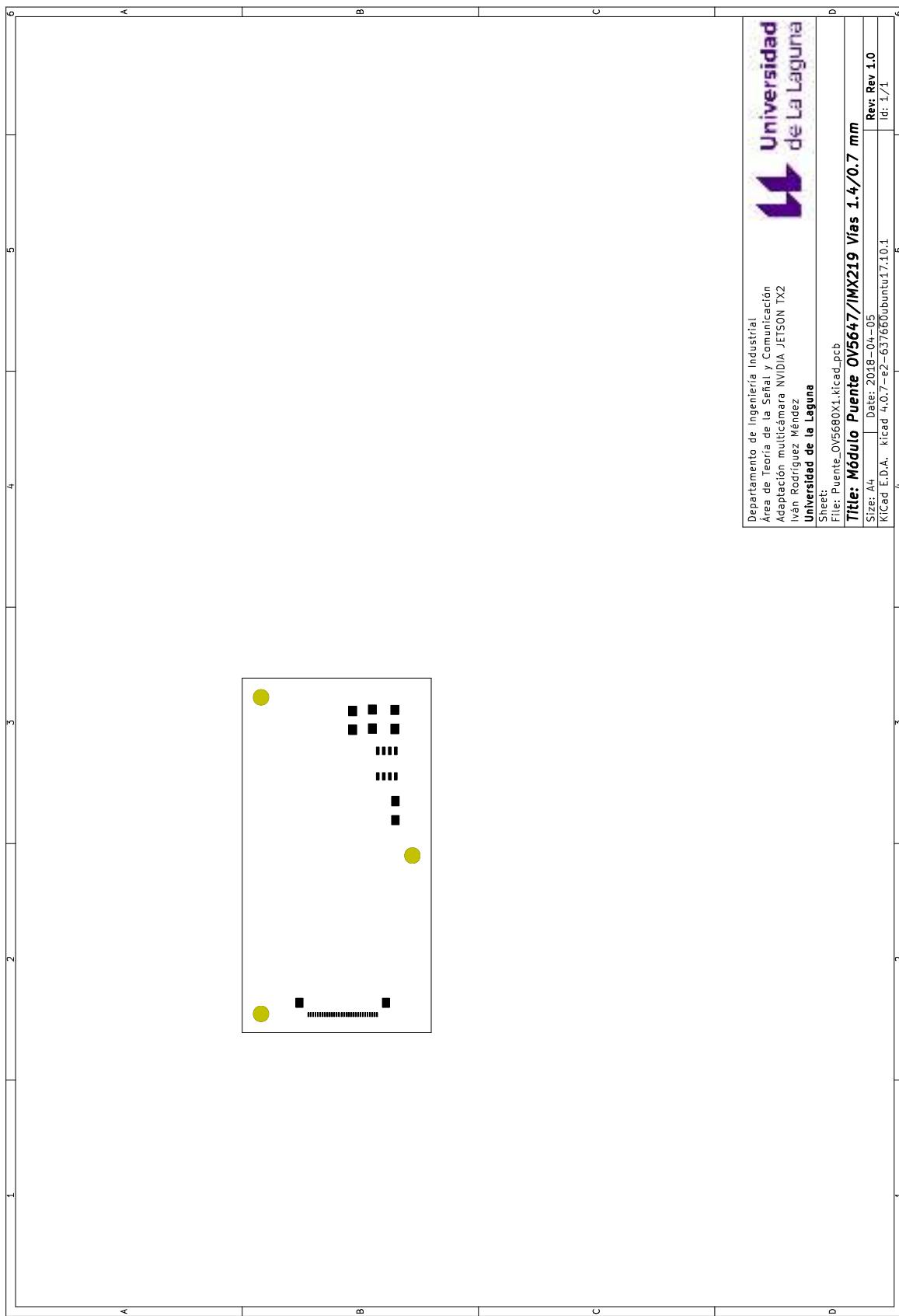
218ANEXO G. PLANOS DE DISEÑO DE PLACAS DE CIRCUITO IMPRESO



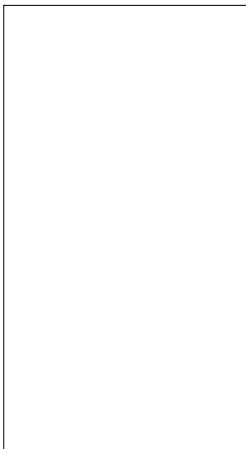


220ANEXO G. PLANOS DE DISEÑO DE PLACAS DE CIRCUITO IMPRESO





222 ANEXO G. PLANOS DE DISEÑO DE PLACAS DE CIRCUITO IMPRESO

| | | | |
|---|---|---|---|
| 1 | | | |
| 2 | | | |
| 3 | | | |
| 4 | | | |
| 5 | | | |
| 6 | | | |
| A | B | C | D |
|  | | | |

Universidad de La Laguna

Departamento de Ingeniería Industrial
Área de Teoría de la Señal y Comunicación
Adaptación multicámaras NVIDIA JETSON TX2
Ivan Rodriguez Vélez
Universidad de la Laguna
Sheet:
File: Puente_0V5647/IMX219_Vías 1.4/0.7 mm
Title: Módulo Puente OV5647/IMX219 Vías 1.4/0.7 mm
Size: A4 Date: 2018-04-05 Rev: Rev 1.0
KICad EDA, kitad 4.0.7-e2-637660ubuntu17.10.1 Id: 1/1

| | | | | | | | | |
|---|---|--|---|--|---|--|---|--|
| | A | | B | | C | | D | |
| 6 | | | | | | | | |
| 5 | | | | | | | | |
| 4 | | | | | | | | |
| 3 | | | | | | | | |
| 2 | | | | | | | | |
| 1 | | | | | | | | |
| | | | | | | | | |

A-0-H-10-00-010

Universidad
de La Laguna

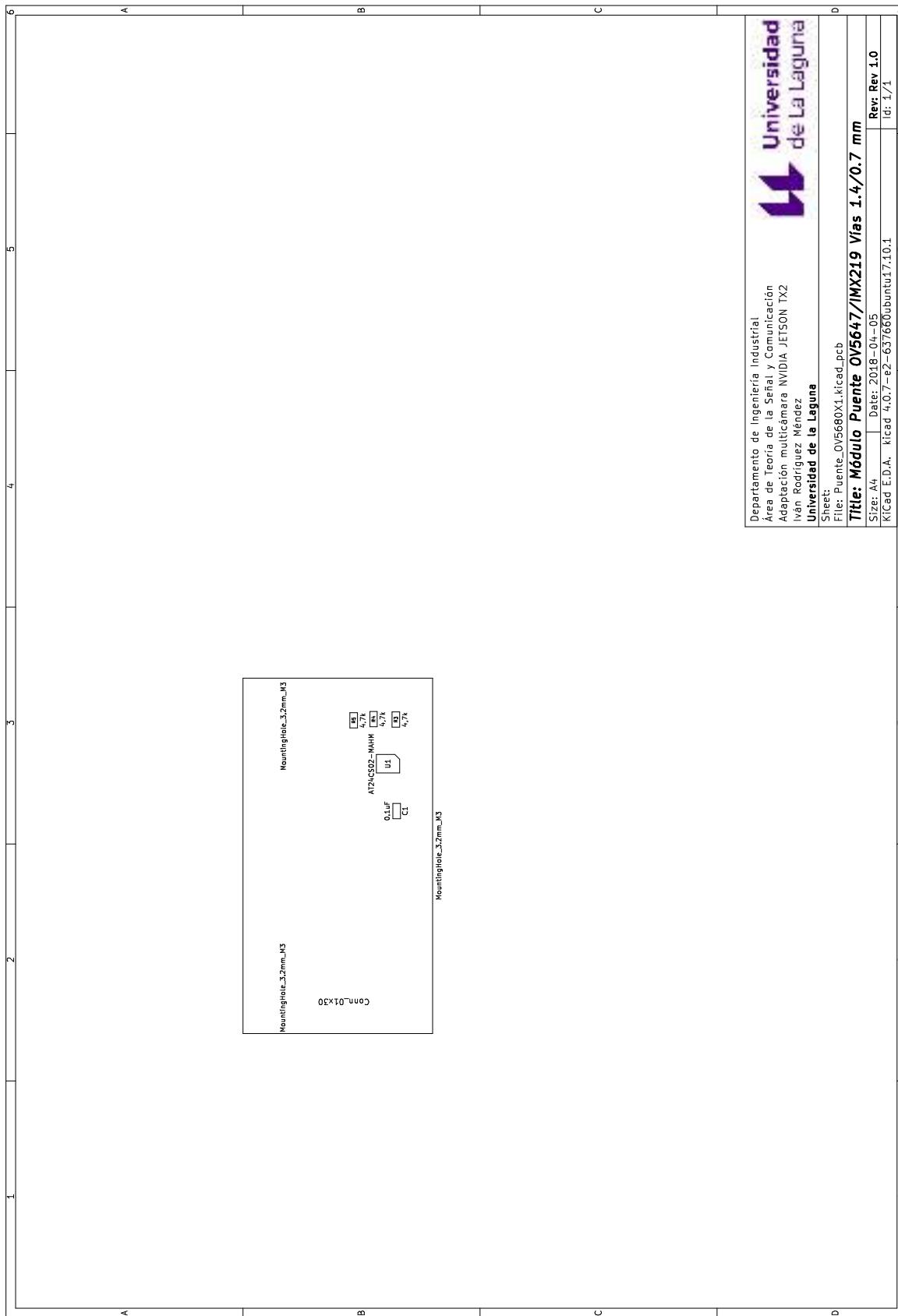
Departamento de Ingeniería Industrial
Área de Teoría de la Señal y Comunicación
Adaptación multicámaras NVIDIA JETSON TX2
Ivan Rodriguez Vélez
Universidad de la Laguna

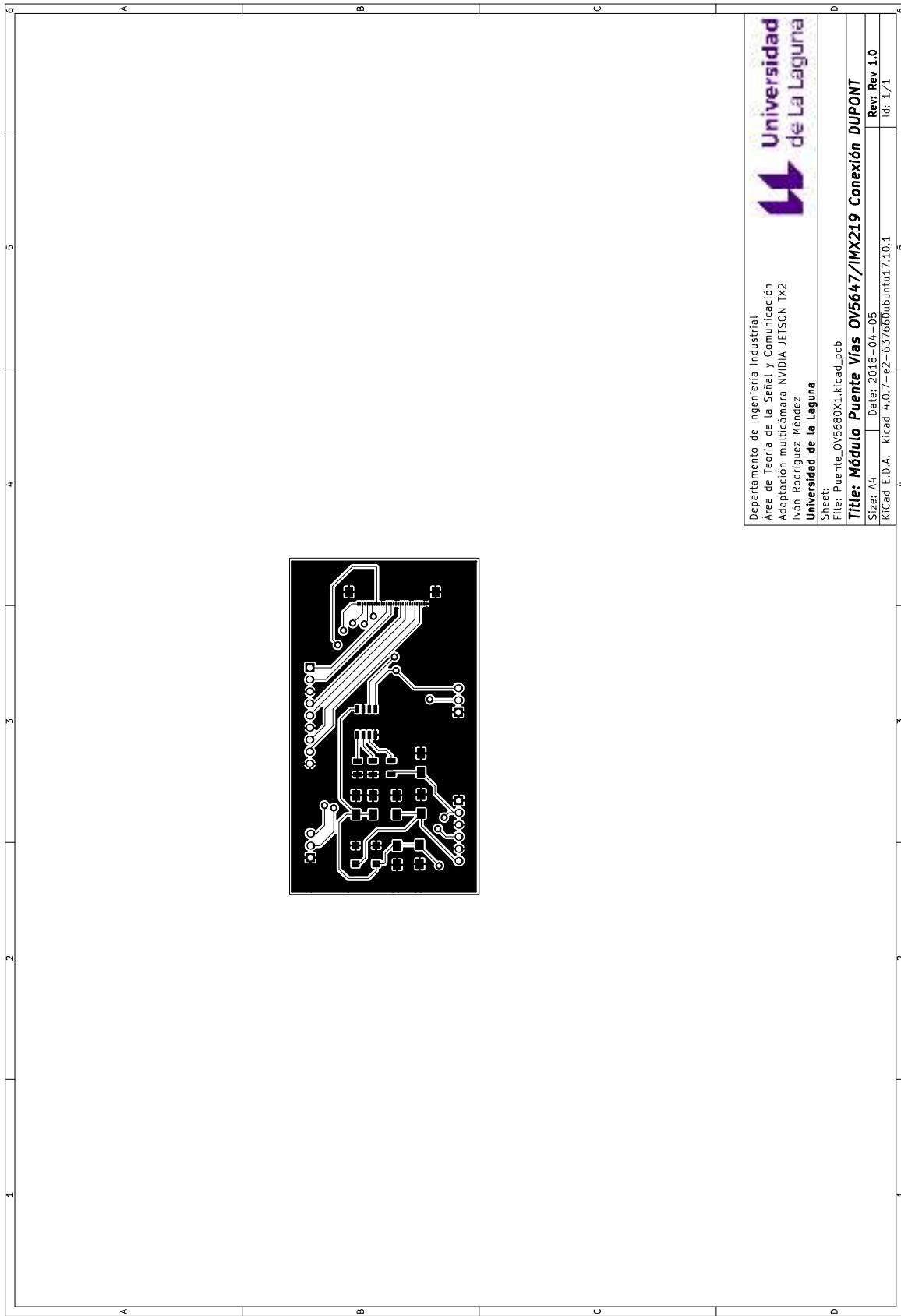
Sheet:
File: Puente_0V5647.ladicad.Lpcb

Title: Módulo Puente 0V5647/IMX219 Vías 1.4/0.7 mm

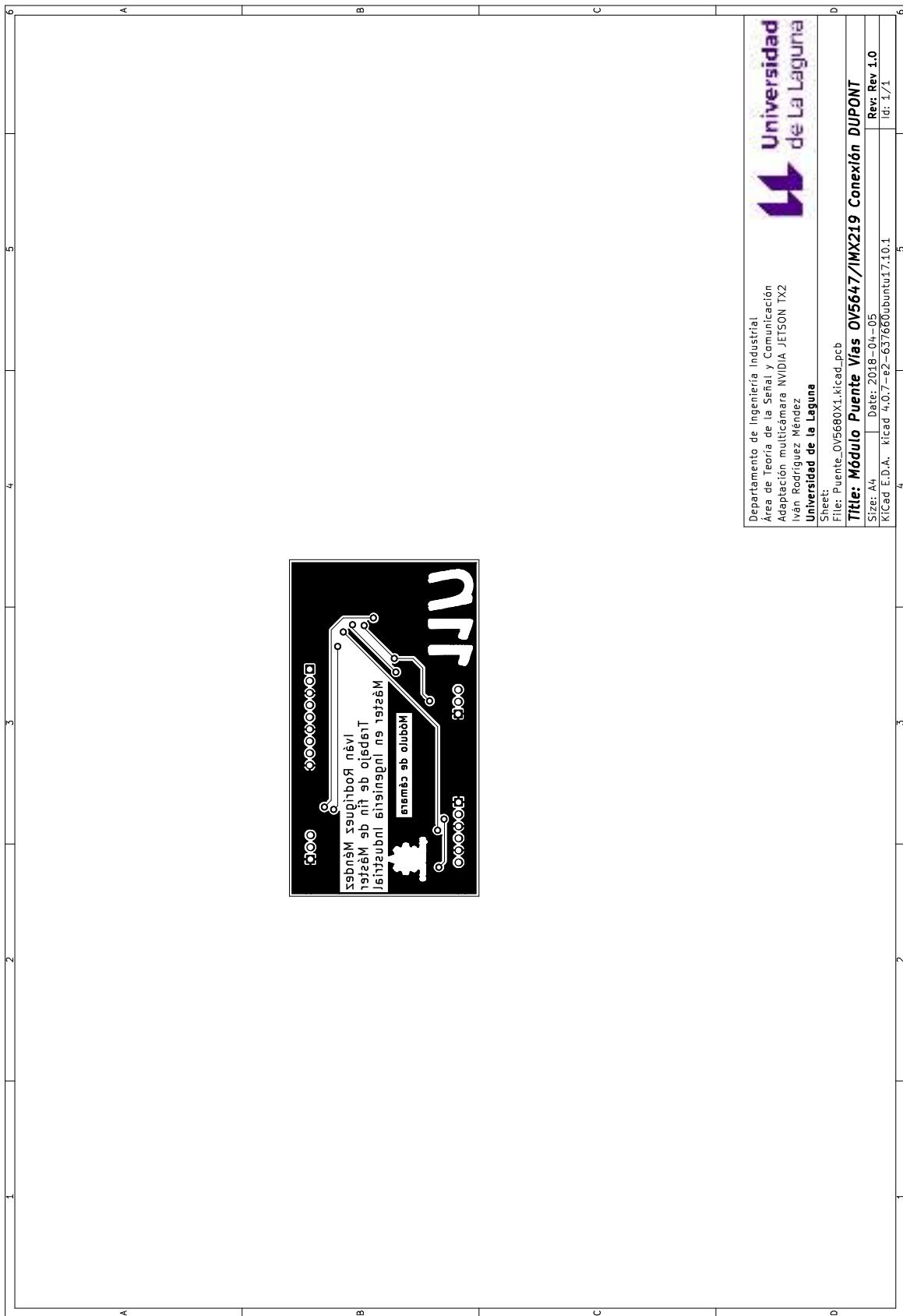
| | | |
|---|------------------|--------------|
| Size: A4 | Date: 2018-04-05 | Rev: Rev 1.0 |
| KICad EDA, kitad 4.0.7-e2-637660ubuntu17.10.1 | | Id: 1/1 |
| 4 | 3 | 2 |
| 1 | | |

224 ANEXO G. PLANOS DE DISEÑO DE PLACAS DE CIRCUITO IMPRESO





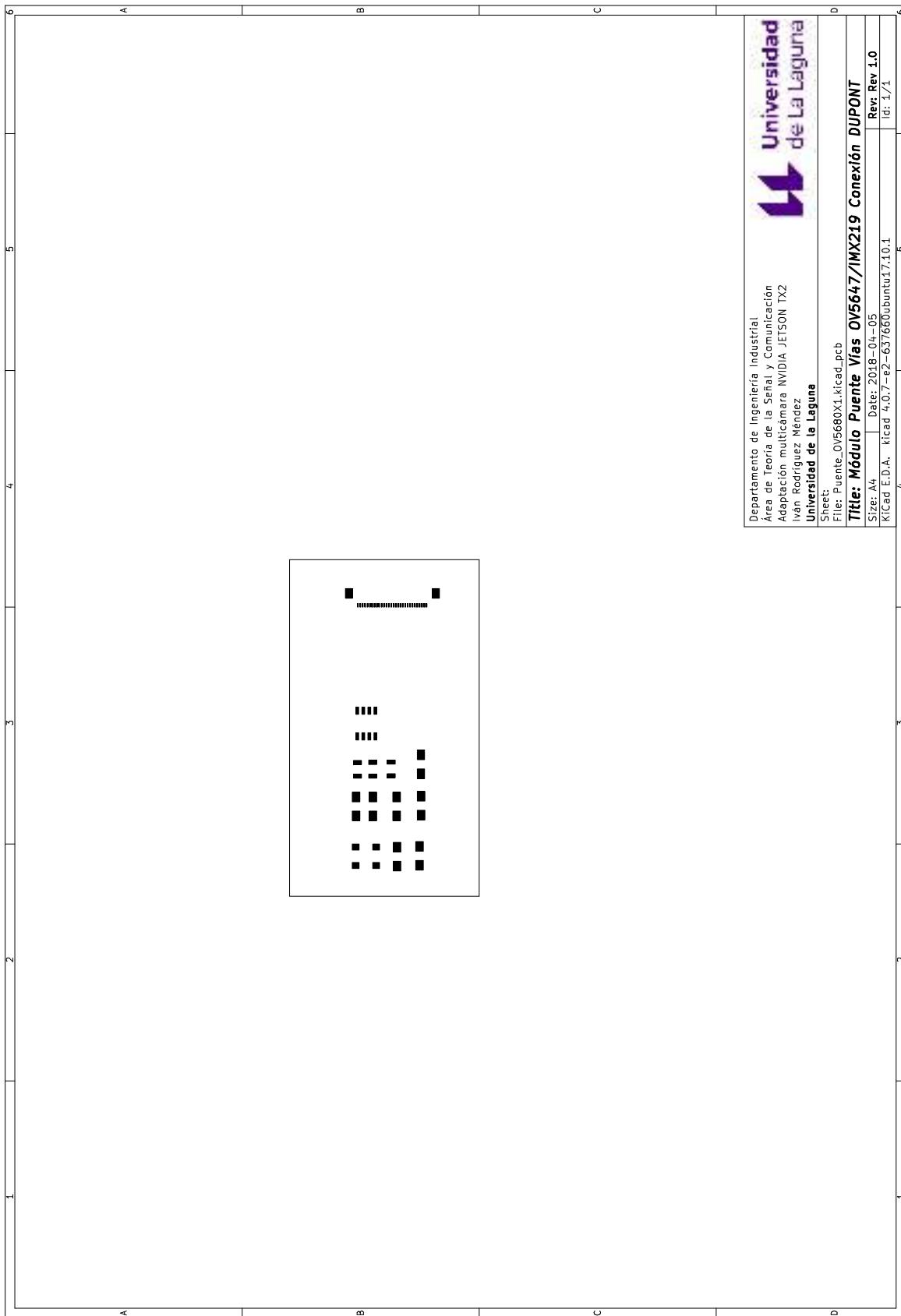
226ANEXO G. PLANOS DE DISEÑO DE PLACAS DE CIRCUITO IMPRESO

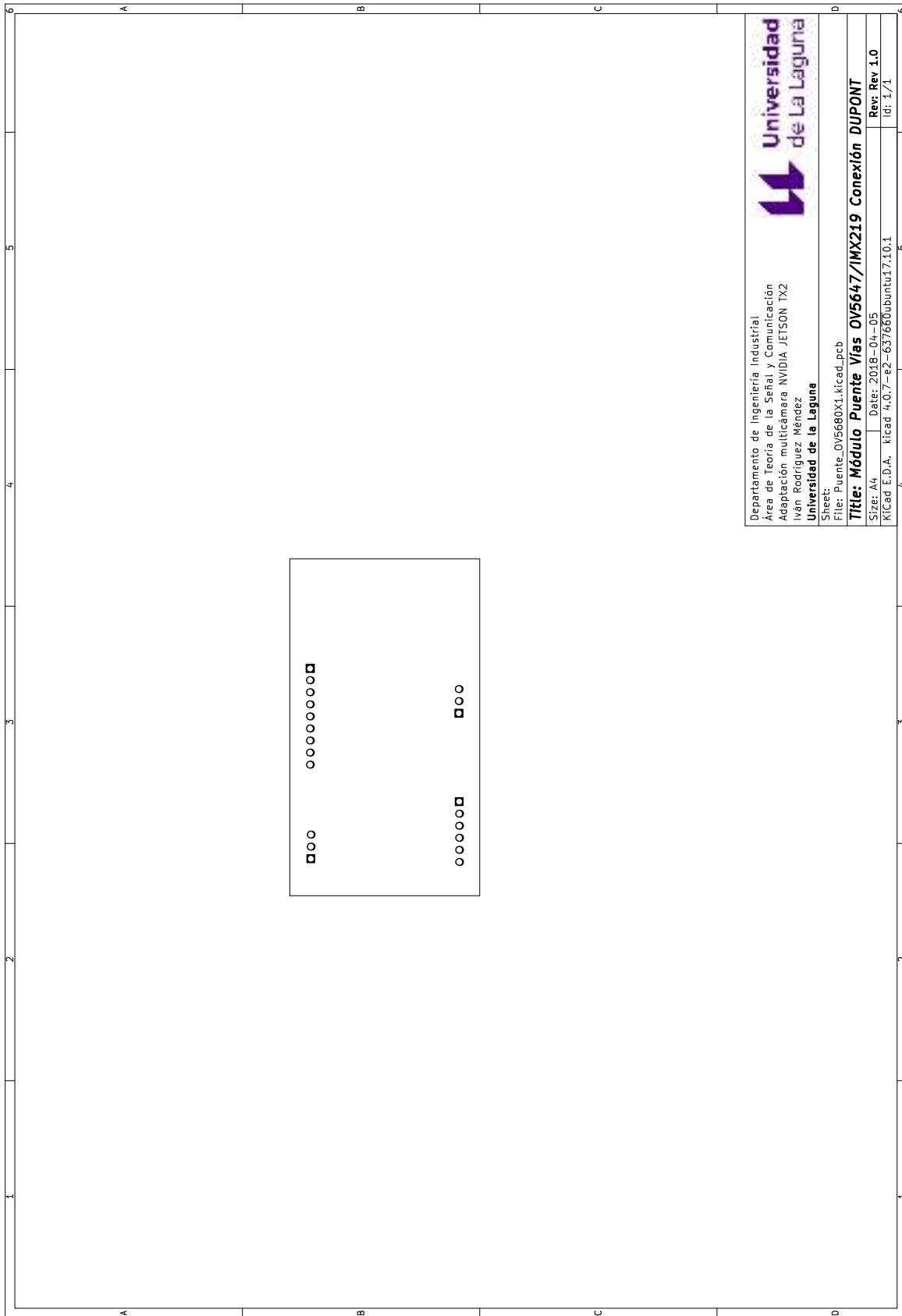


| | | | | |
|---|---|---|---|---|
| | A | B | C | D |
| 6 | | | | |
| 5 | | | | |
| 4 | | | | |
| 3 | | | | |
| 2 | | | | |
| 1 | | | | |
| | | | | |

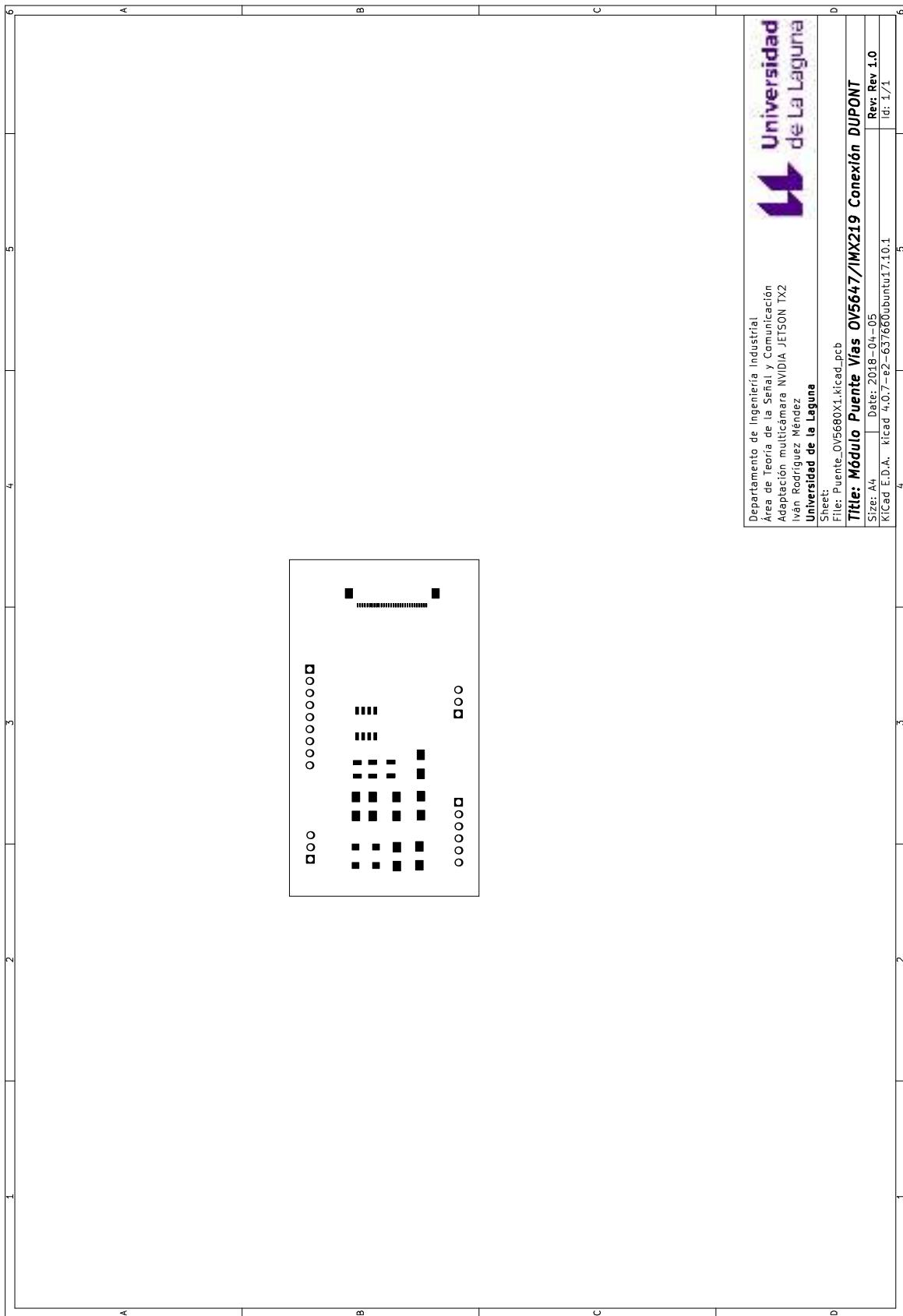
| | | |
|--|---|---|
| Departamento de Ingeniería Industrial Área de Teoría de la Señal y Comunicación Adaptación multicámaras NVIDIA JETSON TX2 Ivan Rodriguez Vélez Universidad de La Laguna |  | Universidad de La Laguna |
| Sheet: File: Puente_015660X4.Vicad.Lpcb | | |
| Title: Módulo Puente Vías Ov5647/IMX219 Conexión DUPONT | Rev: Rev 1.0 | |
| Size: A4 Date: 2018-04-05 | | Id: 1/1 |
| KICad EDA, kitad 4.0.7-e2-637660ubuntu17.10.1 | | |
| 4 | 3 | 2 |
| 3 | 1 | |

228 ANEXO G. PLANOS DE DISEÑO DE PLACAS DE CIRCUITO IMPRESO





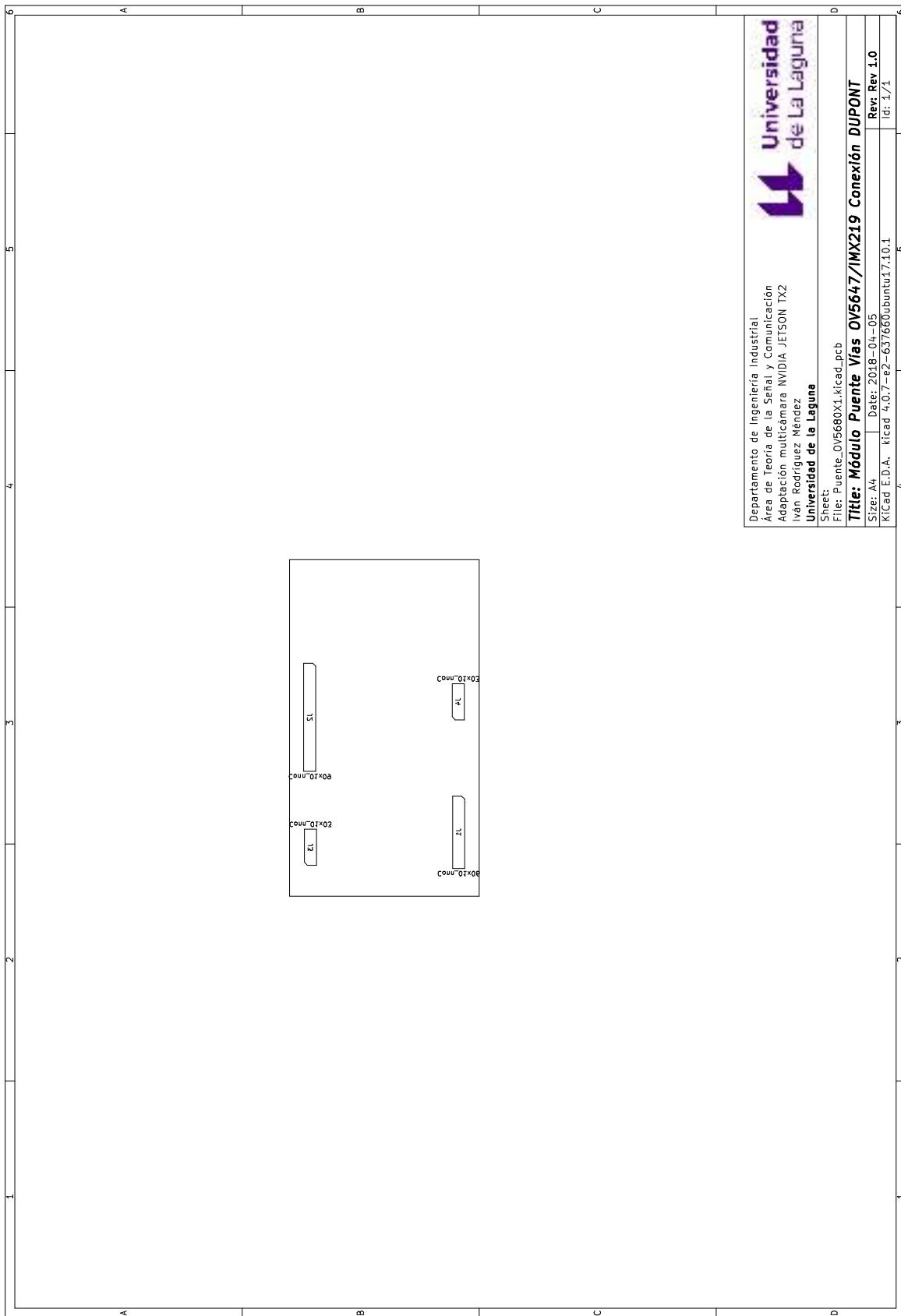
230ANEXO G. PLANOS DE DISEÑO DE PLACAS DE CIRCUITO IMPRESO

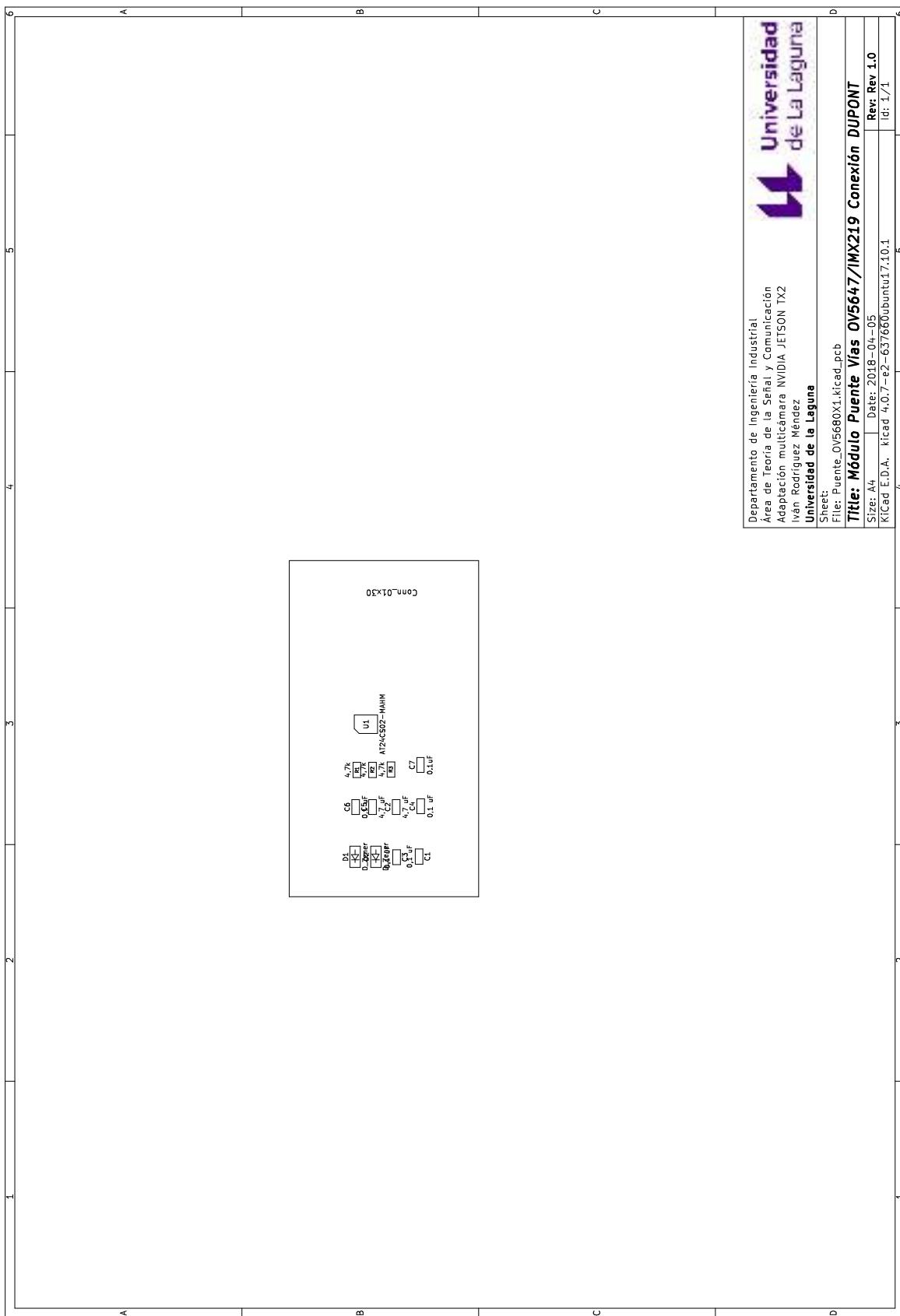


| | | | |
|---|---|---|---|
| A | B | C | D |
| 6 | | | |
| 5 | | | |
| 4 | | | |
| 3 | | | |
| 2 | | | |
| 1 | | | |
| | | | |
| A | B | C | D |
| 6 | | | |
| 5 | | | |
| 4 | | | |
| 3 | | | |
| 2 | | | |
| 1 | | | |

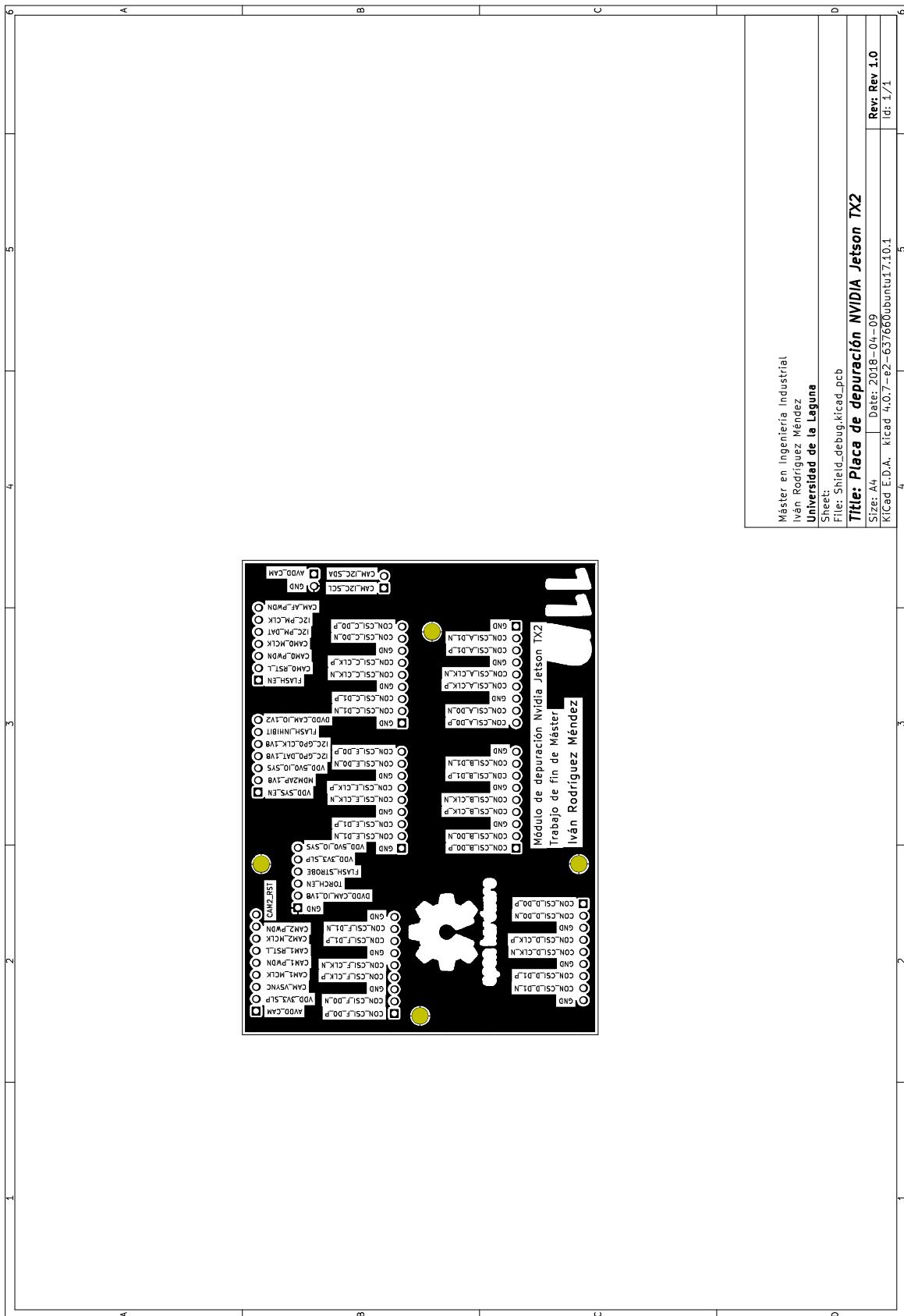
Departamento de Ingeniería Industrial
Área de Teoría de la Señal y Comunicación
Adaptación multicámaras NVIDIA JETSON TX2
Ivan Rodriguez Vélez
Universidad de La Laguna
Sheet:
File: Puente_015660X4.Licad.Lpcb
Title: Módulo Puente Vías Ov5647/IMX219 Conexión DUPONT
Size: A4 Date: 2018-04-05 Rev: Rev 1.0
KICad EDA, kitad 4.0.7-e2-637660ubuntu17.10.1 Id: 1/1

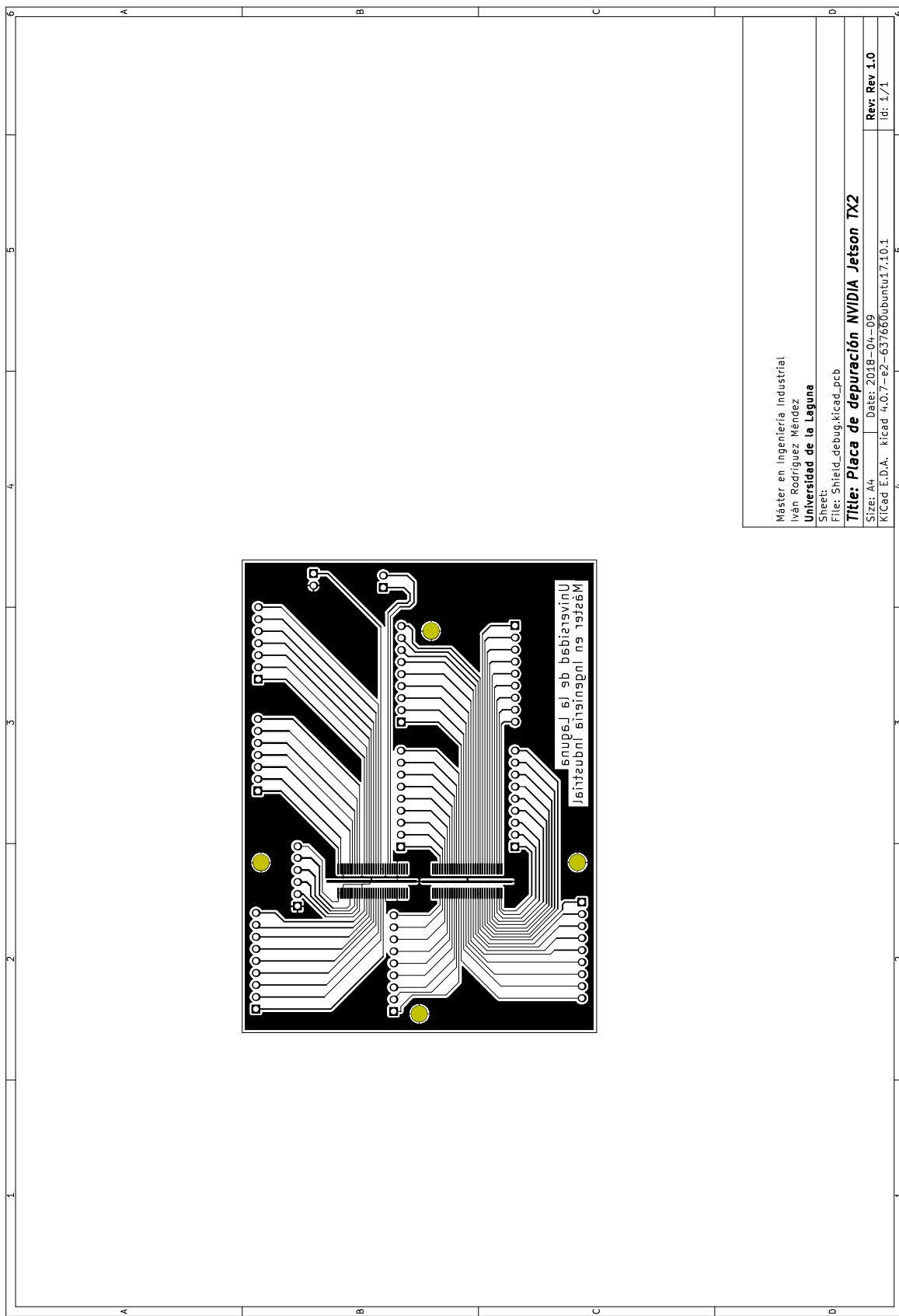
232 ANEXO G. PLANOS DE DISEÑO DE PLACAS DE CIRCUITO IMPRESO



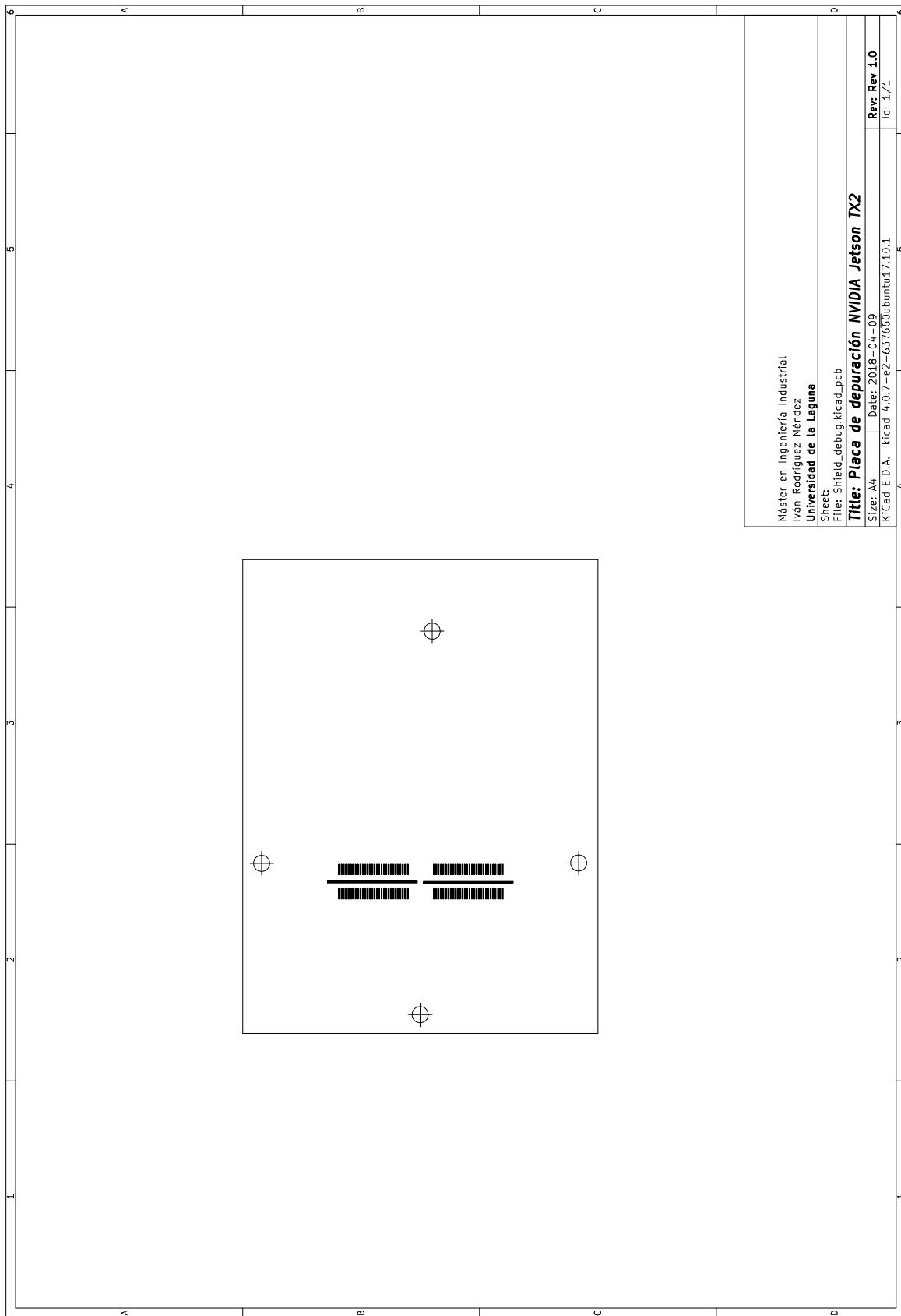


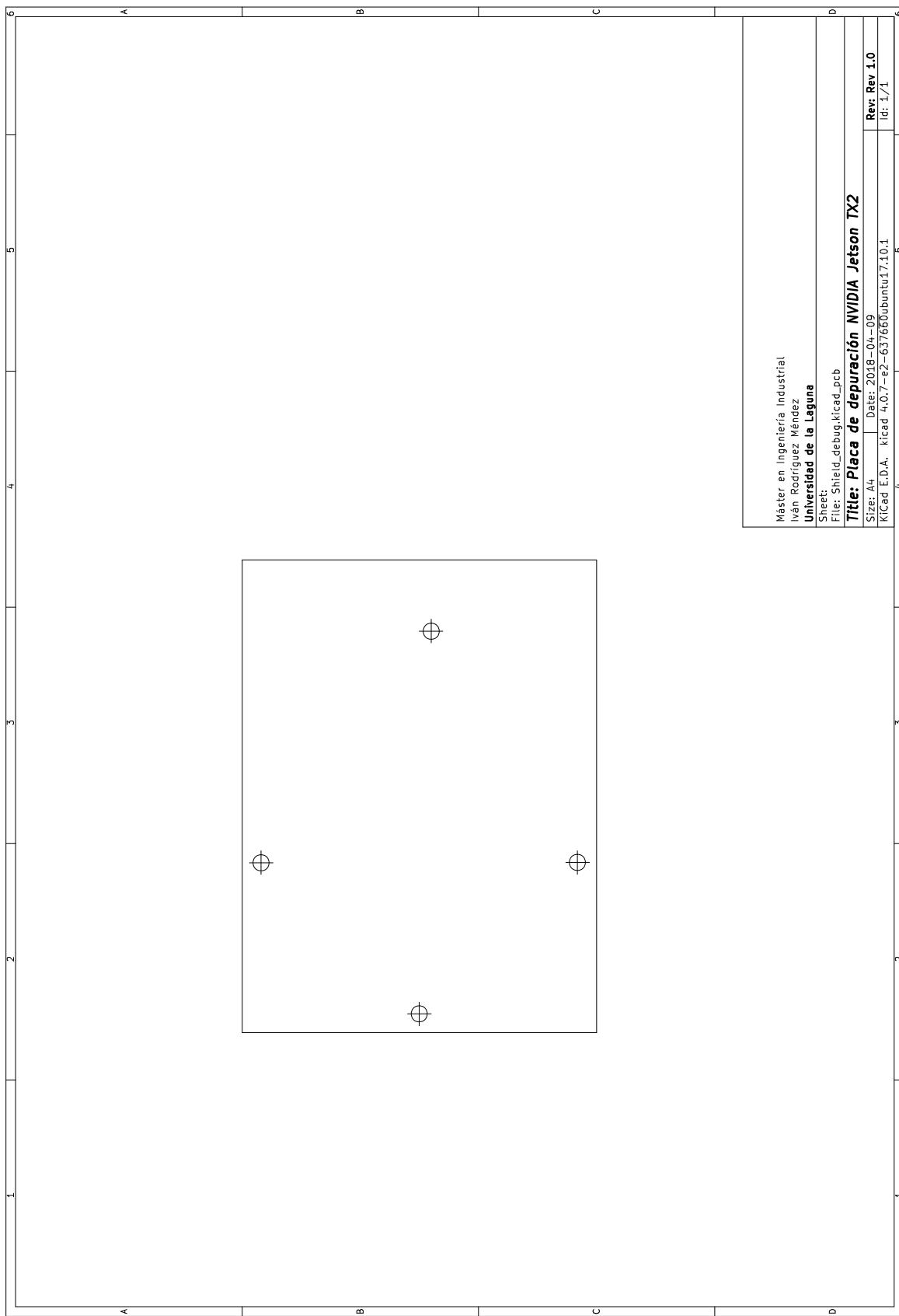
234 ANEXO G. PLANOS DE DISEÑO DE PLACAS DE CIRCUITO IMPRESO



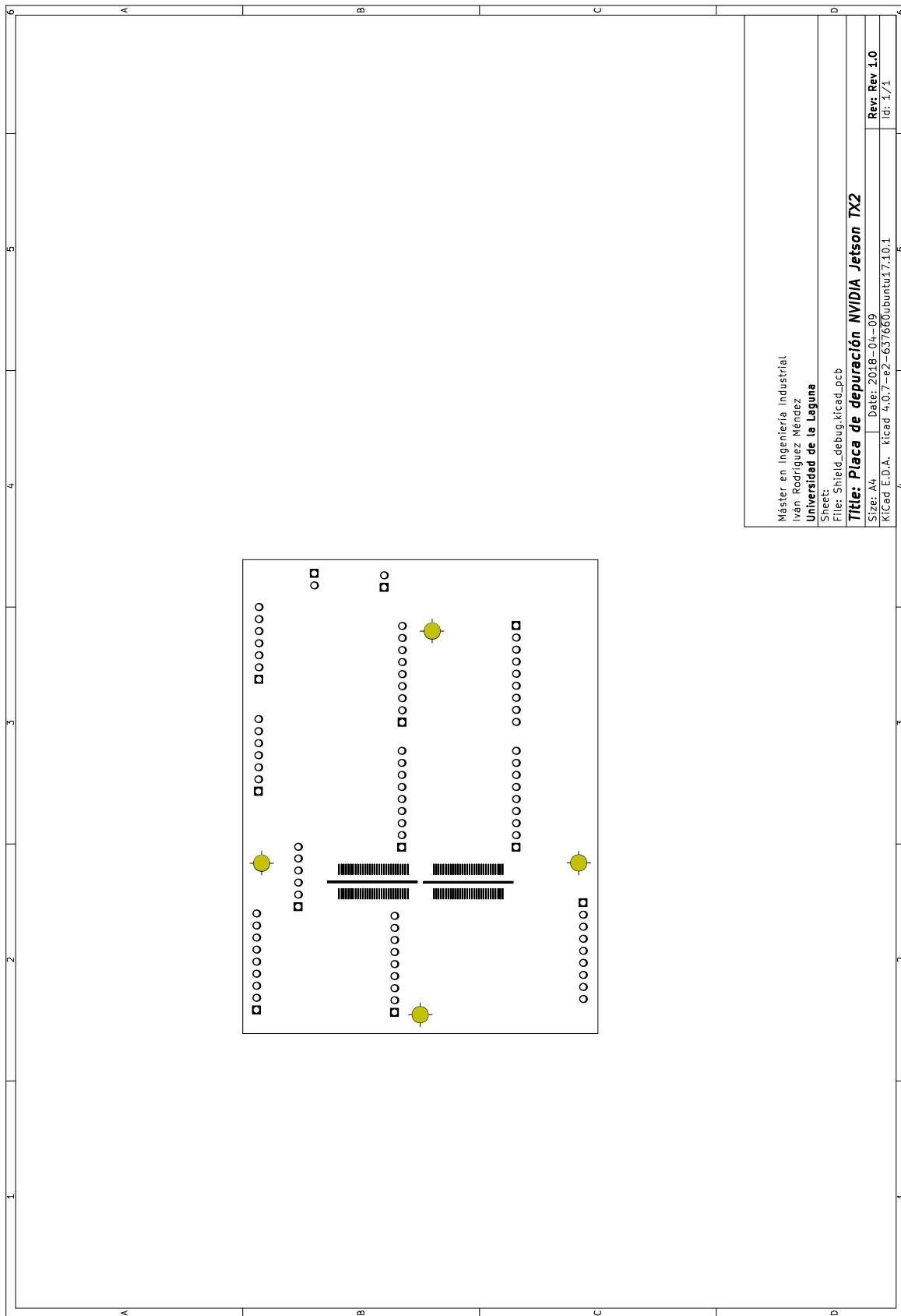


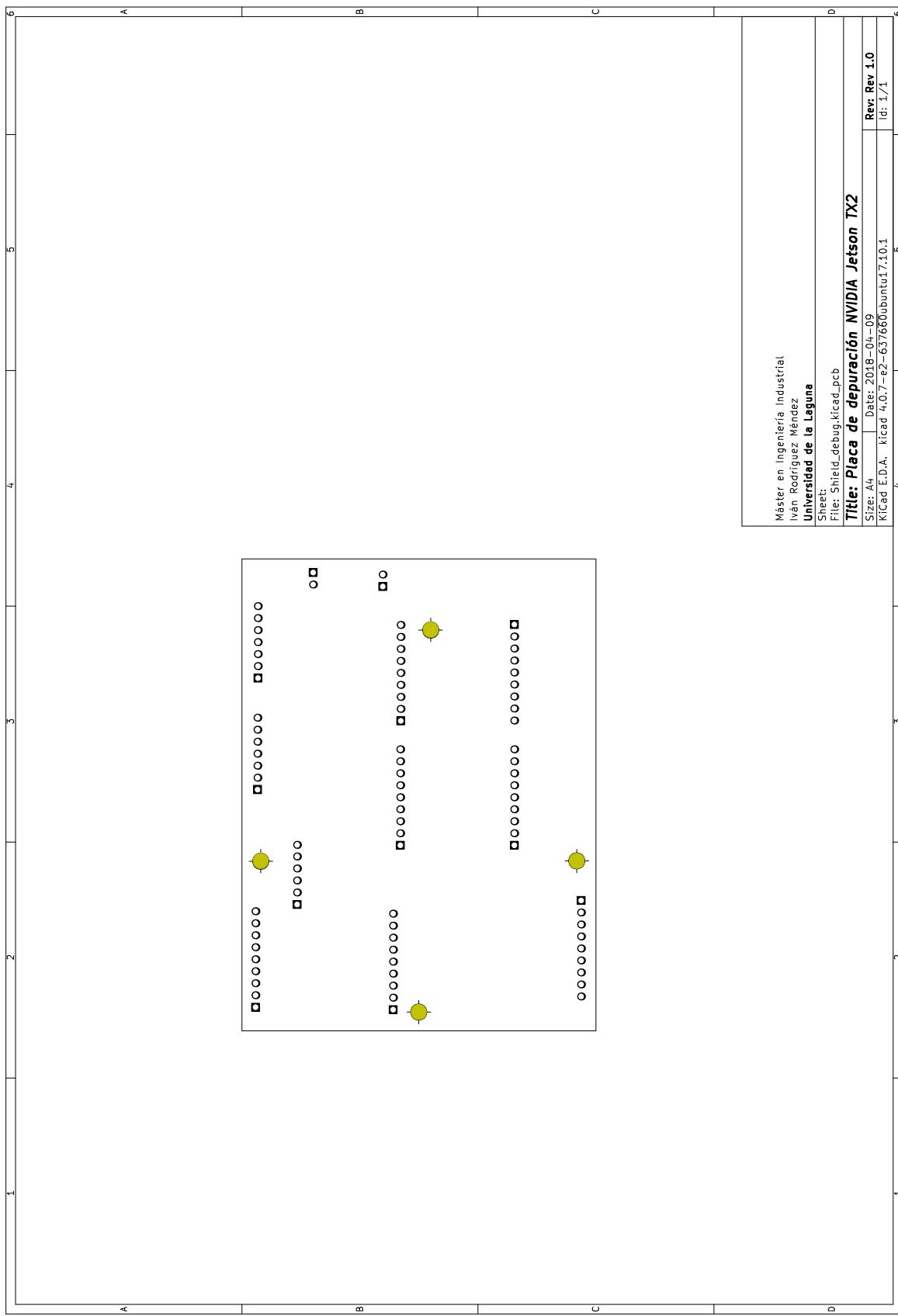
236ANEXO G. PLANOS DE DISEÑO DE PLACAS DE CIRCUITO IMPRESO



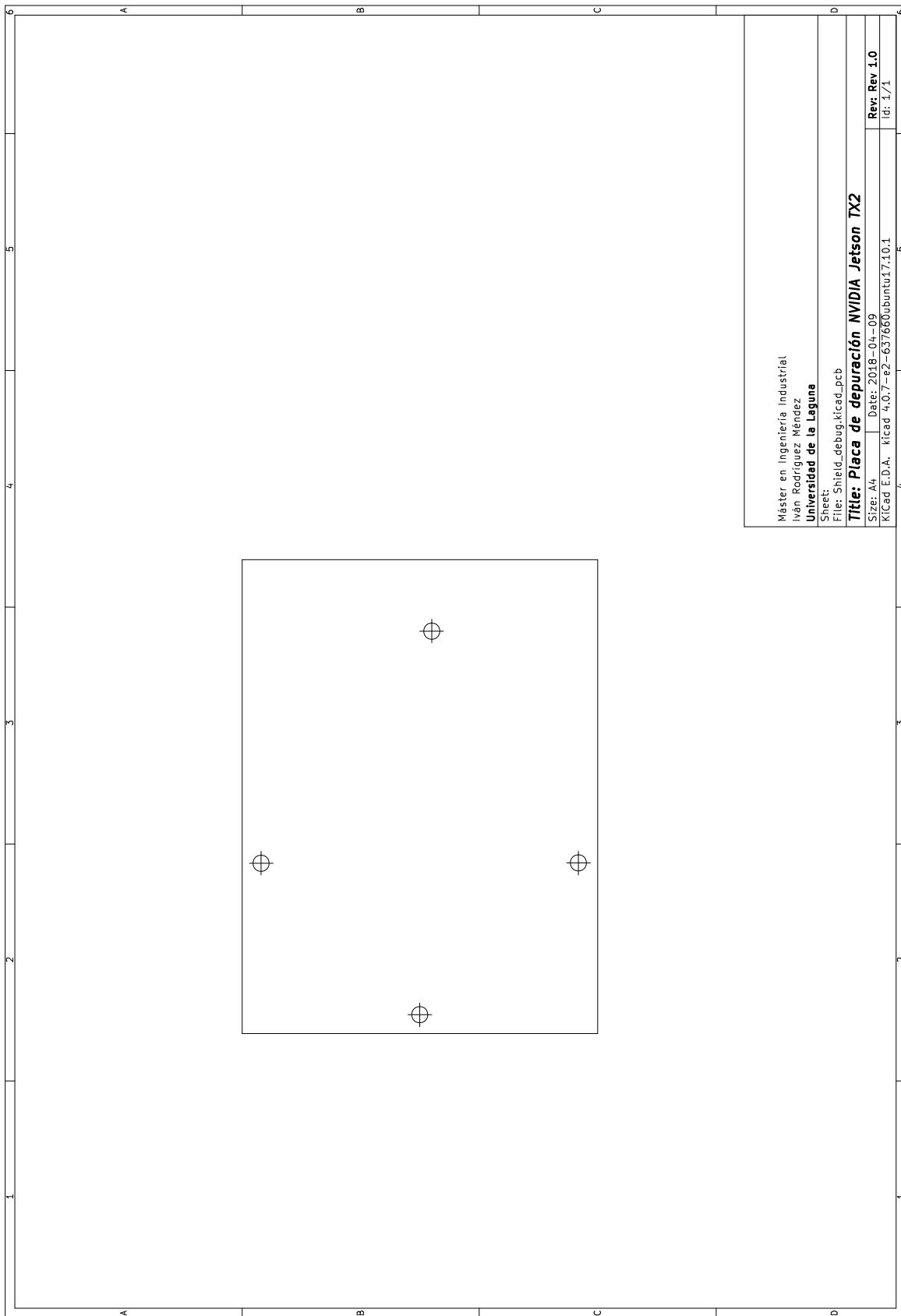


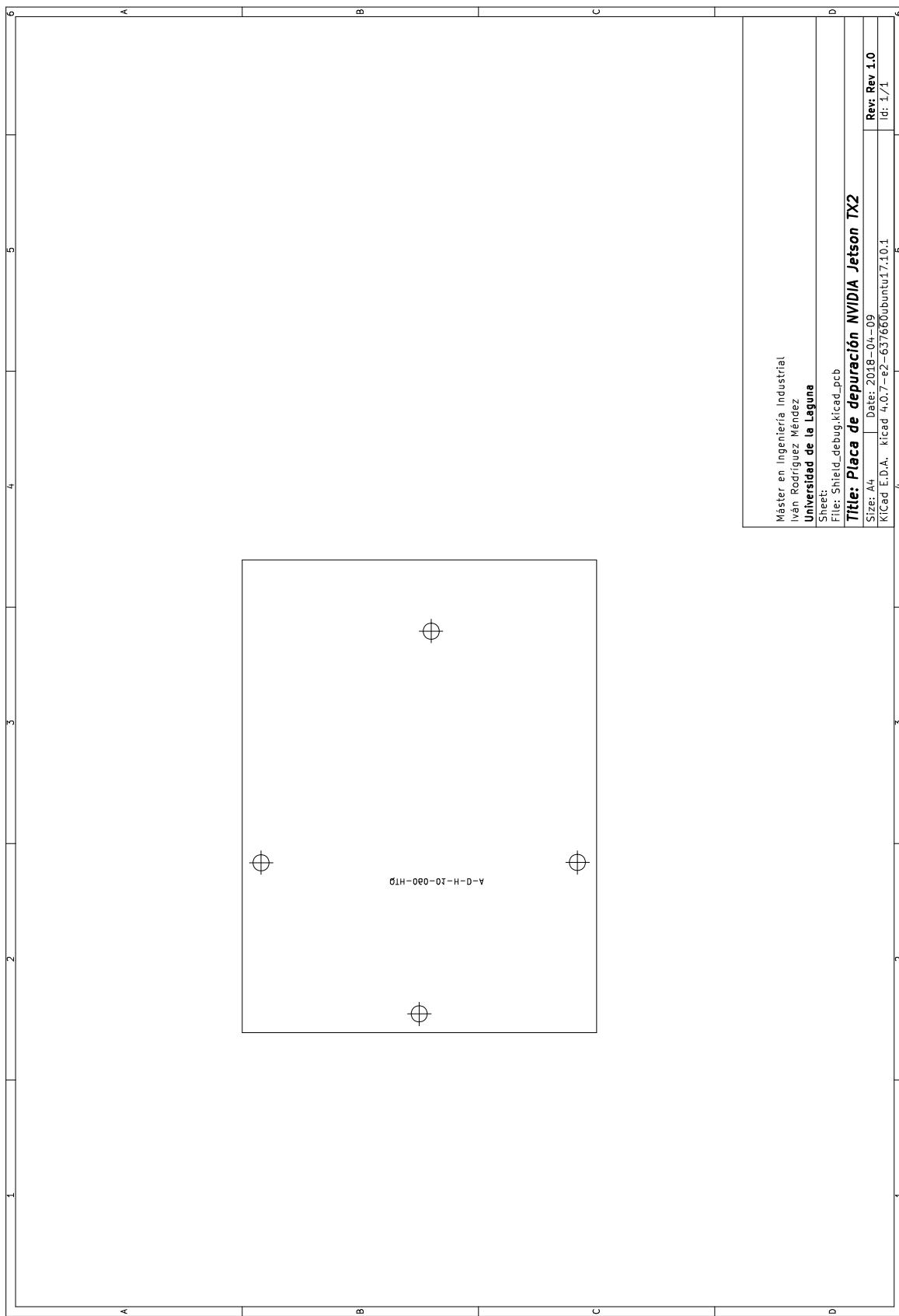
238 ANEXO G. PLANOS DE DISEÑO DE PLACAS DE CIRCUITO IMPRESO



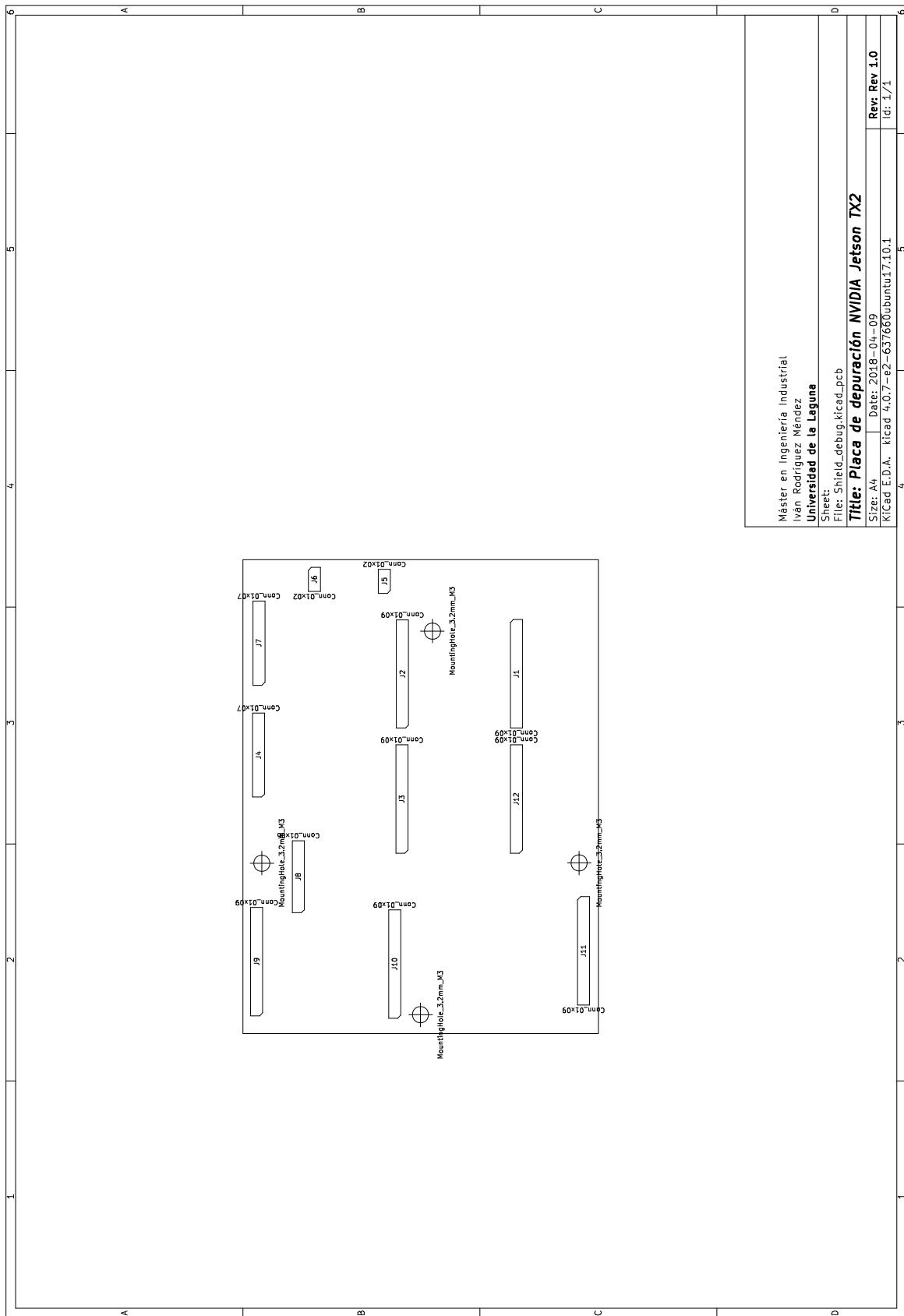


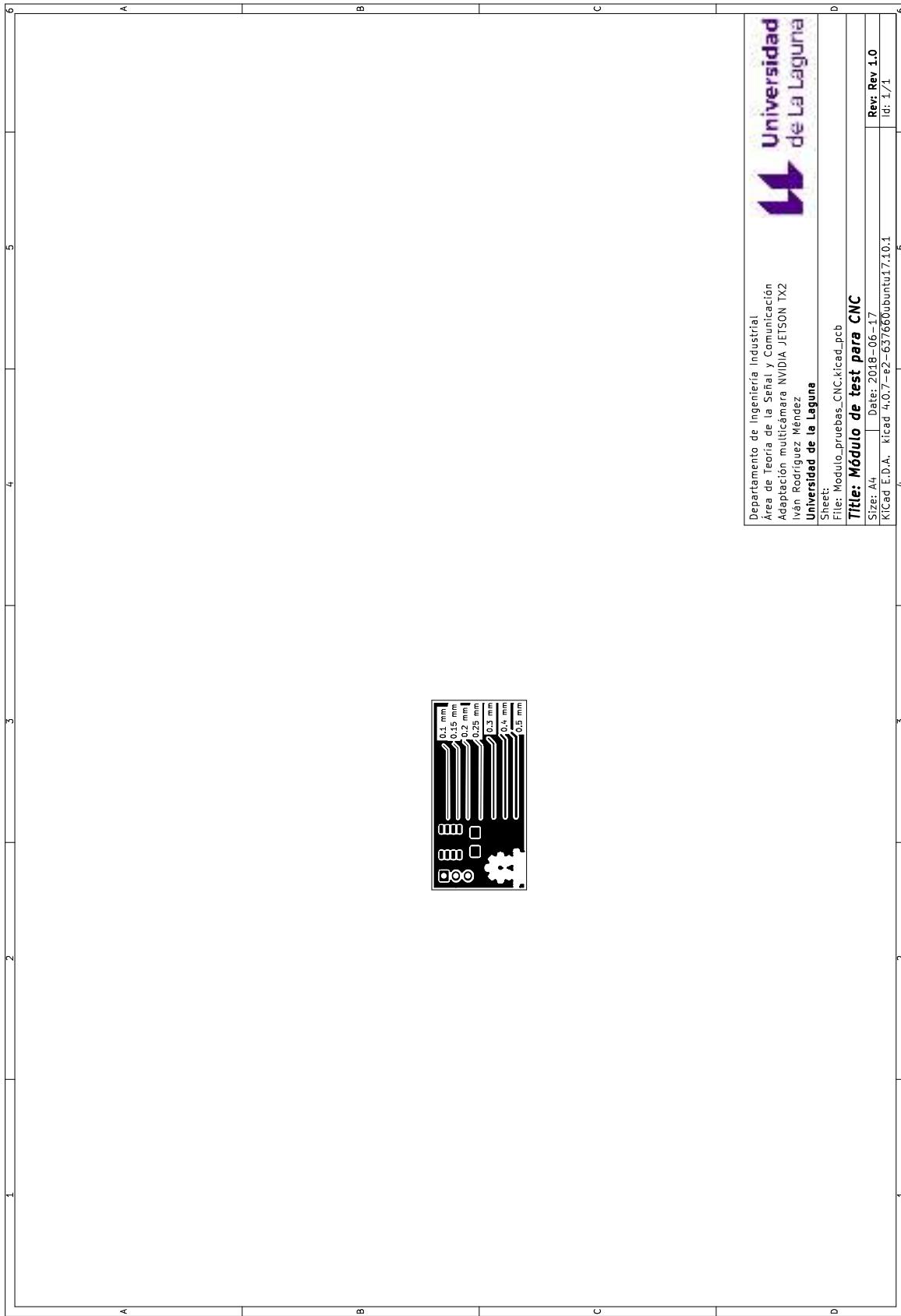
240 ANEXO G. PLANOS DE DISEÑO DE PLACAS DE CIRCUITO IMPRESO





242 ANEXO G. PLANOS DE DISEÑO DE PLACAS DE CIRCUITO IMPRESO





244 ANEXO G. PLANOS DE DISEÑO DE PLACAS DE CIRCUITO IMPRESO

Anexo H

Modelos 3D de los módulos desarrollados

En este anexo se recogen los modelos 3D de los diferentes módulos que se han diseñado. El orden de presentación de los modelos es el siguiente:

1. Modelos 3D del módulo de cámara para el sensor OV5680.
2. Modelos 3D del módulo puente para el sensor OV5680.
3. Modelos 3D del módulo puente para el sensor IMX219/OV5647.
4. Modelos 3D del módulo puente para el sensor IMX219/OV5647 con conexión dupont.
5. Modelos 3D del módulo de depuración
6. Modelos 3D del módulo de pruebas para CNC.

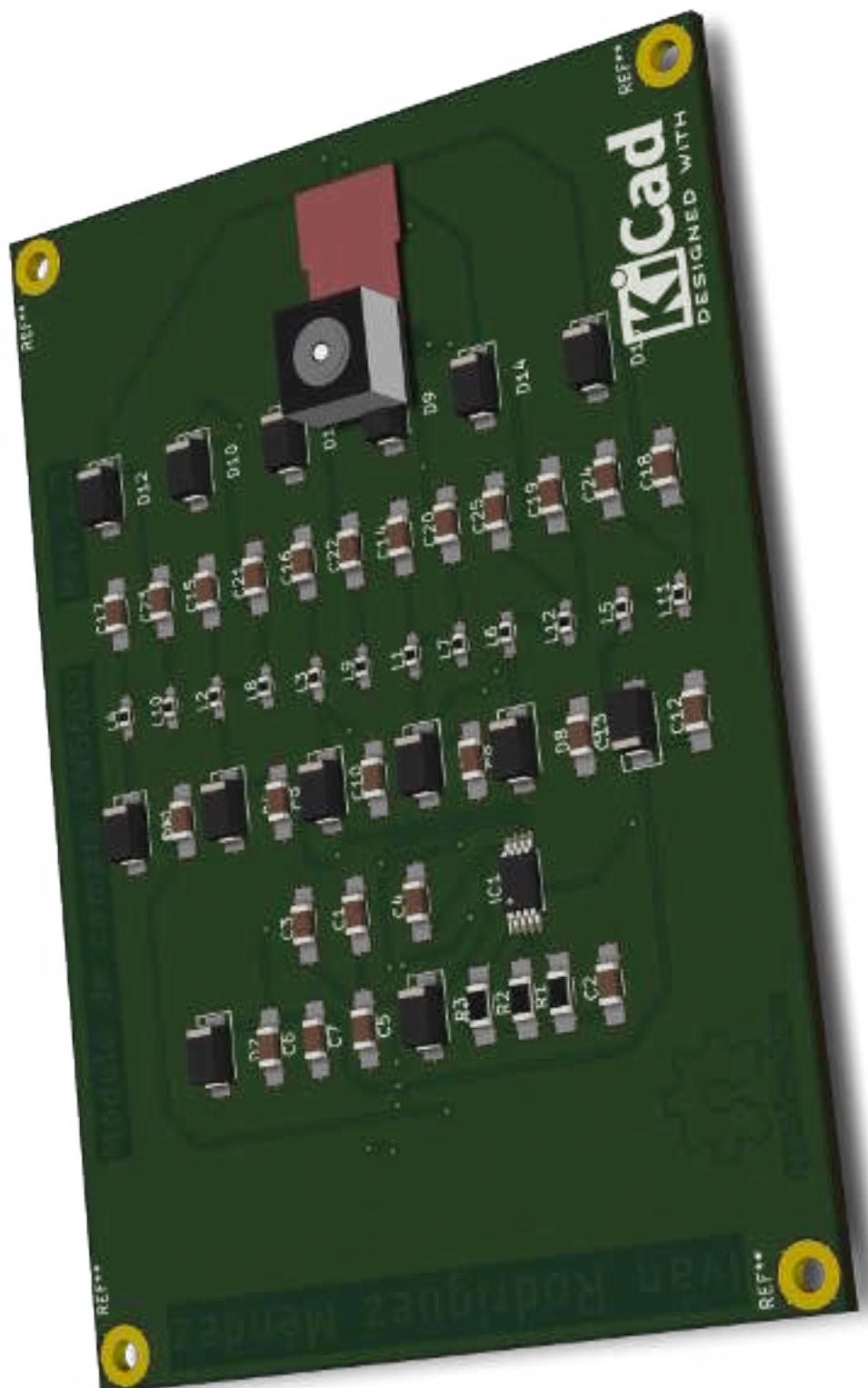


Figura H.1: Visualización 3D del módulo de cámara para el sensor OV5680



Figura H.2: Visualización 3D del módulo de cámara para el sensor OV5680



Figura H.3: Visualización 3D del módulo puente para el sensor OV5680

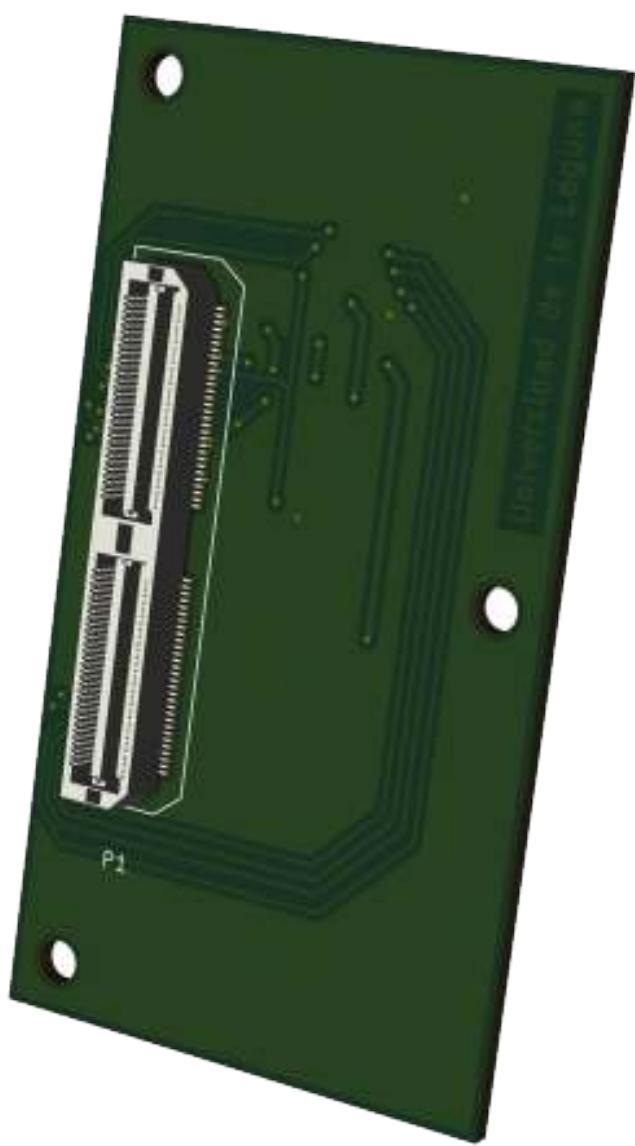


Figura H.4: Visualización 3D del módulo puente para el sensor OV5680

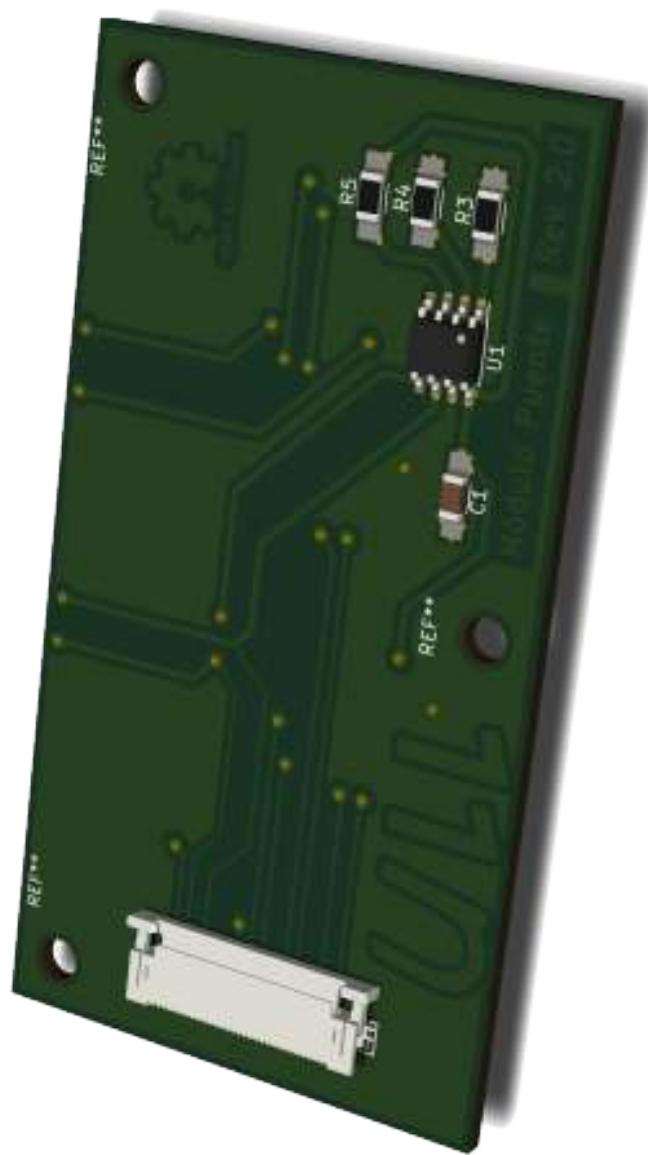


Figura H.5: Visualización 3D del módulo puente para el sensor IMX219/OV5647



Figura H.6: Visualización 3D del módulo puente para el sensor IMX219/OV5647

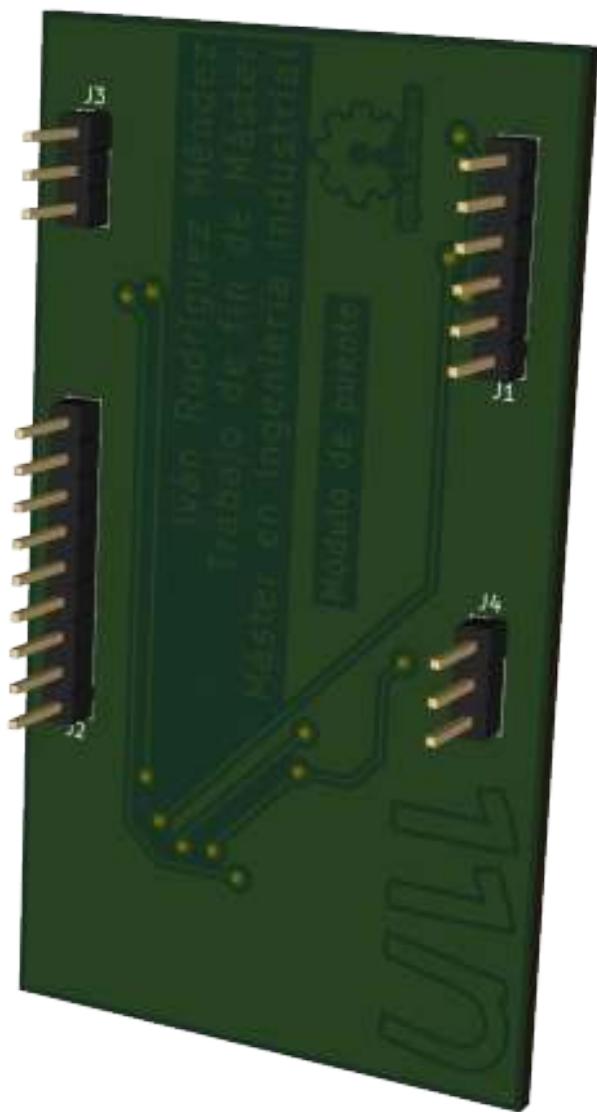


Figura H.7: Visualización 3D del módulo puente para el sensor IMX219/OV5647 con conexión dupont

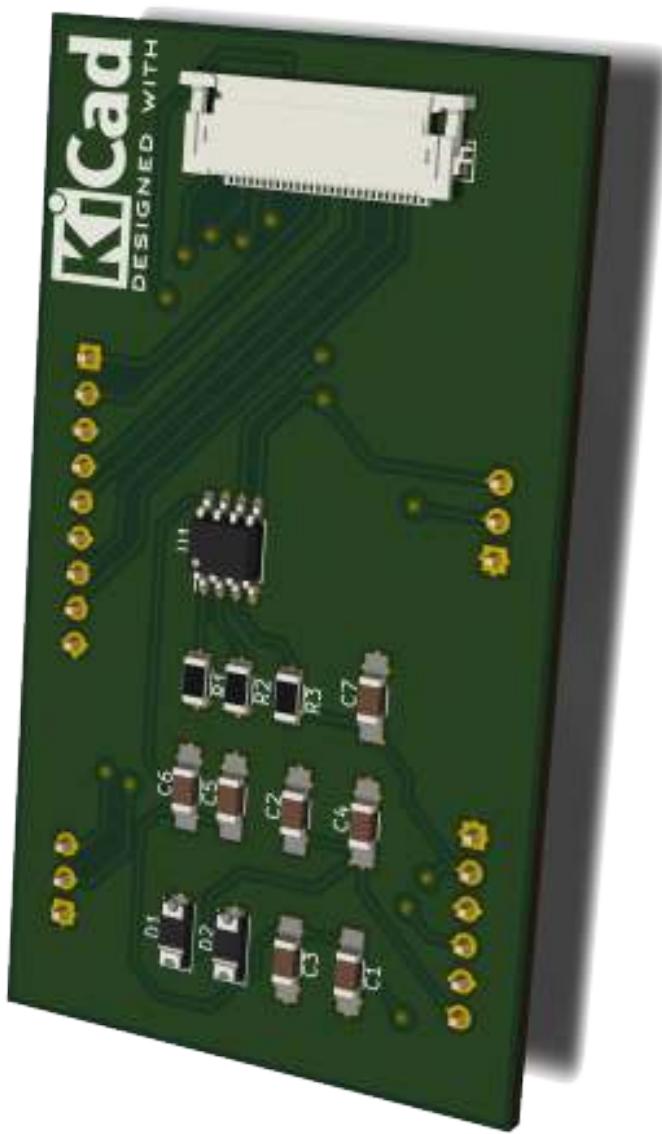


Figura H.8: Visualización 3D del módulo puente para el sensor IMX219/OV5647 con conexión dupont

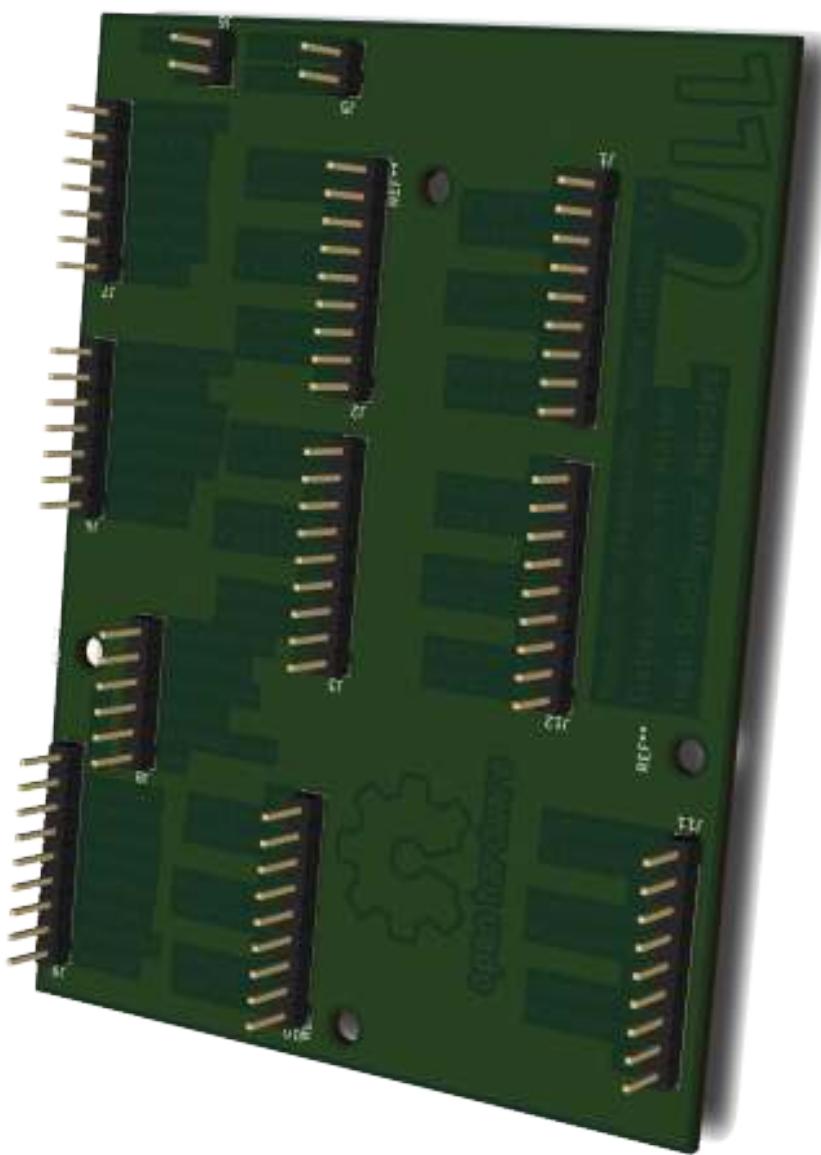


Figura H.9: Visualización 3D del módulo de depuración

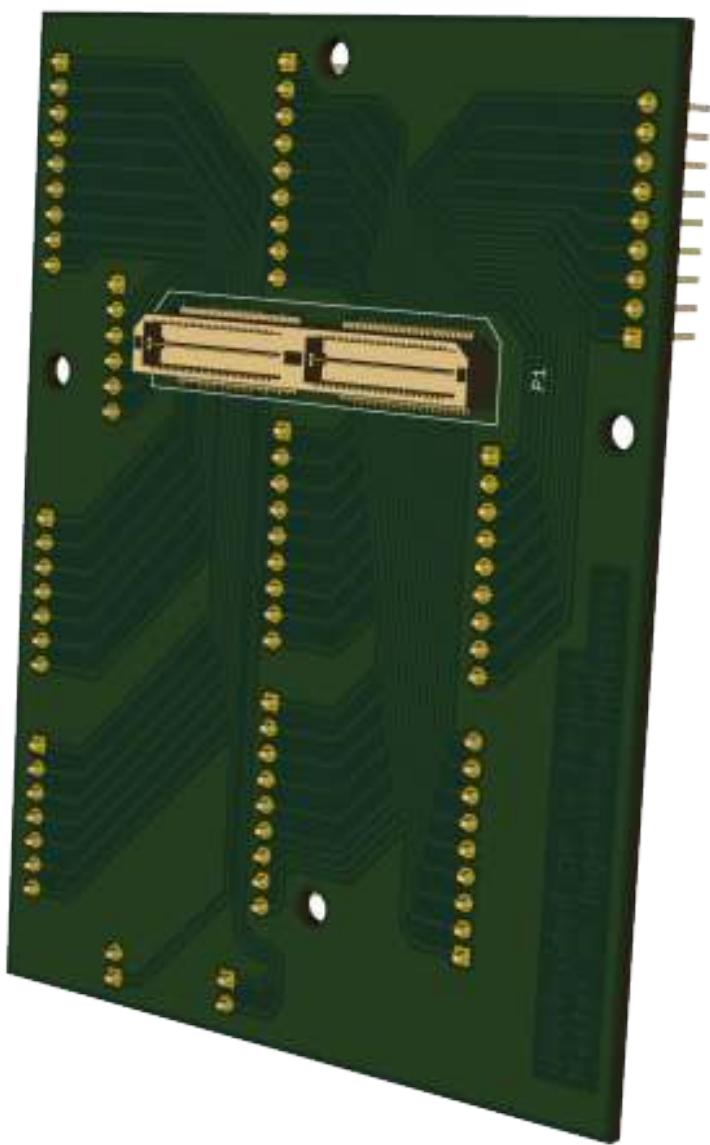


Figura H.10: Visualización 3D del módulo de depuración

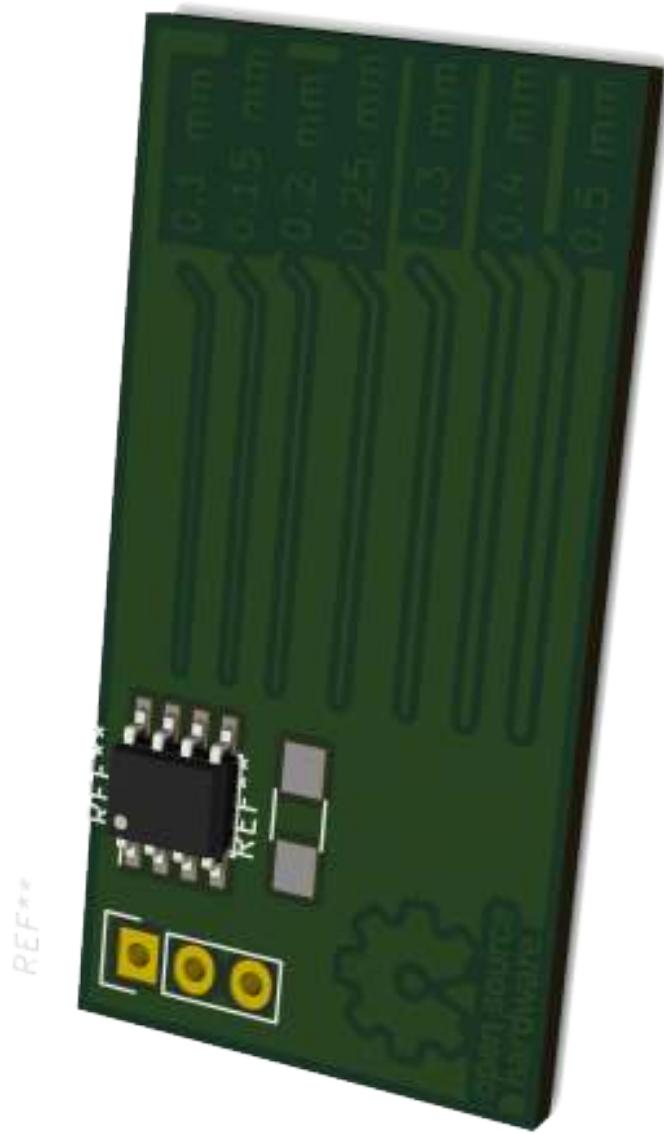


Figura H.11: Visualización 3D módulo de pruebas para CNC

Anexo I

Características de los componentes

En este anexo se recogen los componentes necesarios para la fabricación de las placas de circuito impreso. Se presentarán en diferentes secciones tablas que muestran la información de los componentes necesarios, así como el tipo de encapsulado que tienen, su valor y la cantidad de cada uno de ellos que se requiere. El orden de las secciones dentro de este anexo es el siguiente:

1. Componentes para la placa del sensor de cámara OV5680
2. Componentes para el módulo puente sensor OV5680
3. Componentes módulo puente para el sensor IMX219/OV5647
4. Componentes módulo puente con conexión DUPONT
5. Componentes módulo puente con conexión DUPONT sin EEPROM
6. Componentes módulo de depuración de líneas

I.1. Componentes para el módulo del sensor de cámara OV5680

En la tabla I.1, podemos ver los componentes necesarios para la fabricación del módulo de cámara basado en el sensor OV5680.

| Grupo | Componente | Valor | Encapsulado | Unidades |
|---------------|---------------------|-------------|-------------|----------|
| Condensadores | C1206C104J5RACTU | 100nF | 1206 (in) | 5 |
| | C3216X7R1E475K160AC | 4.7 μ F | 1206 (in) | 2 |
| | C1206C120J5GACTU | 12pF | 1206 (in) | 18 |

Tabla I.1 – continua en la página siguiente

Tabla I.1 – continuación de la página anterior

| Grupo | Componente | Valor | Encapsulado | Unidades |
|------------------|-----------------|----------------------|------------------------------------|----------|
| Módulo de cámara | OV5680 | - | - | 1 |
| Resistencias | CRG1206F4K7 | 4.7kΩF | 1206 (in) | 3 |
| Conecotor | Molex | Serie FFC/FPC 30 PIN | SMT | 1 |
| Bobinas | HK212518NK-T | 19nH | 0603 (in) | 12 |
| Diodos ZE-NER | 1SMA5917BT3G | 1.5 W 2.5μA 5Ω | SMA (in) | 14 |
| EEPROM | AT24C02D-SSHM-T | EEPROM 1.7V | TSSOP-8 4.4x3mm Pitch 0.65mm | 1 |

Tabla I.1: Características de los componentes del módulo de cámara OV5680

I.2. Componentes para el módulo puente sensor OV5680

En la tabla I.2, podemos ver los componentes necesarios para la fabricación del módulo puente para de cámara basado en el sensor OV5680.

| Grupo | Componente | Valor | Encapsulado | Unidades |
|---------------|------------------|----------------------|------------------------------------|----------|
| Condensadores | C1206C104J5RACTU | 100nF | 1206 (in) | 1 |
| Resistencias | CRG1206F4K7 | 4.7kΩF | 1206 (in) | 3 |
| Conecotor | Molex | Serie FFC/FPC 30 PIN | SMT | 1 |
| Conecotor | SAMTEC | QTH-060-01-L-D-A | SMT | 1 |
| EEPROM | AT24C02D-SSHM-T | EEPROM 1.7V | TSSOP-8 4.4x3mm Pitch 0.65mm | 1 |

Tabla I.2: Características de los componentes del módulo puente para el sensor OV5680

I.3. Componentes módulo puente para el sensor IMX219/OV5647

En la tabla I.3, podemos ver los componentes necesarios para la fabricación del módulo puente para de cámara basado en los sensores OV5647/IMX219.

| Grupo | Componente | Valor | Encapsulado | Unidades |
|---------------|------------------|----------------------|-------------|----------|
| Condensadores | C1206C104J5RACTU | 100nF | 1206 (in) | 1 |
| Resistencias | CRG1206F4K7 | 4.7kΩF | 1206 (in) | 3 |
| Conecotor | Molex | Serie FFC/FPC 30 PIN | SMT | 1 |
| Conecotor | SAMTEC | QTH-060-01-L-D-A | SMT | 1 |

Tabla I.3 – continua en la página siguiente

I.5. COMPONENTES MÓDULO PUENTE CON CONEXIÓN DUPONT SIN EEPROM259

Tabla I.3 – continuación de la página anterior

| Grupo | Componente | Valor | Encapsulado | Unidades |
|--------|-----------------|-------------|------------------------------------|----------|
| EEPROM | AT24C02D-SSHM-T | EEPROM 1.7V | TSSOP-8 4.4x3mm Pitch 0.65mm | 1 |

Tabla I.3: Características de los componentes del módulo puente para el sensor OV5647

I.4. Componentes módulo puente con conexión DUPONT

En la tabla I.4, podemos ver los componentes necesarios para la fabricación del módulo puente para de cámara basado en los sensores OV5647/IMX219 con conexión de pines rectos para cables tipo DUPONT.

| Grupo | Componente | Valor | Encapsulado | Unidades |
|---------------|------------------|----------------------|------------------------------------|----------|
| Condensadores | C1206C104J5RACTU | 100nF | 1206 (in) | 1 |
| Resistencias | CRG1206F4K7 | 4.7kΩF | 1206 (in) | 3 |
| Conector | Molex | Serie FFC/FPC 30 PIN | SMT | 1 |
| Conector | HARWIN | Recto 1 fila | THD | 1 |
| EEPROM | AT24C02D-SSHM-T | EEPROM 1.7V | TSSOP-8 4.4x3mm Pitch 0.65mm | 1 |

Tabla I.4: Características de los componentes del módulo puente para el sensor OV5647/IMX219 con conexión DUPONT

I.5. Componentes módulo puente con conexión DUPONT sin EEPROM

En la tabla I.5, podemos ver los componentes necesarios para la fabricación del módulo puente para de cámara basado en los sensores OV5647/IMX219 con conexión de pines rectos para cables tipo DUPONT y sin EEPROM. En el caso de los conectores HARWIN se indica una sola unidad en la tabla, sin embargo hay que colocar tantos pines como agujeros de montaje existan. En este caso son necesarios 21 pines.

| Grupo | Componente | Valor | Encapsulado | Unidades |
|---------------|---------------------|-------|-------------|----------|
| Condensadores | C1206C104J5RACTU | 100nF | 1206 (in) | 4 |
| Condensadores | C3216X7R1E475K160AC | 4.7μF | 3216 (in) | 2 |

Tabla I.5 – continua en la página siguiente

Tabla I.5 – *continuación de la página anterior*

| Grupo | Componente | Valor | Encapsulado | Unidades |
|---------------|--------------|----------------------|-------------|----------|
| Diodos ZE-NER | 1SMA5917BT3G | 1.5 W 2.5 μ A 5Ω | SMA (in) | 2 |
| Conector | Molex | Serie FFC/FPC 30 PIN | SMT | 1 |
| Conector | HARWIN | Recto 1 fila | THD | 1 |

Tabla I.5: Características de los componentes del módulo puente para el sensor OV5647/IMX219 con conexión DUPONT sin EEPROM

I.6. Componentes módulo de depuración de líneas

En la tabla I.5, podemos ver los componentes necesarios para la fabricación del módulo de depuración de líneas. En el caso de los conectores *HARWIN* se indica una sola unidad en la tabla, sin embargo hay que colocar tantos pines como agujeros de montaje existan. En este caso son necesarios 87 pines.

| Grupo | Componente | Valor | Encapsulado | Unidades |
|----------|------------|------------------|-------------|----------|
| Conector | SAMTEC | QTH-060-01-L-D-A | SMT | 1 |
| Conector | HARWIN | Recto 1 fila | THD | 1 |

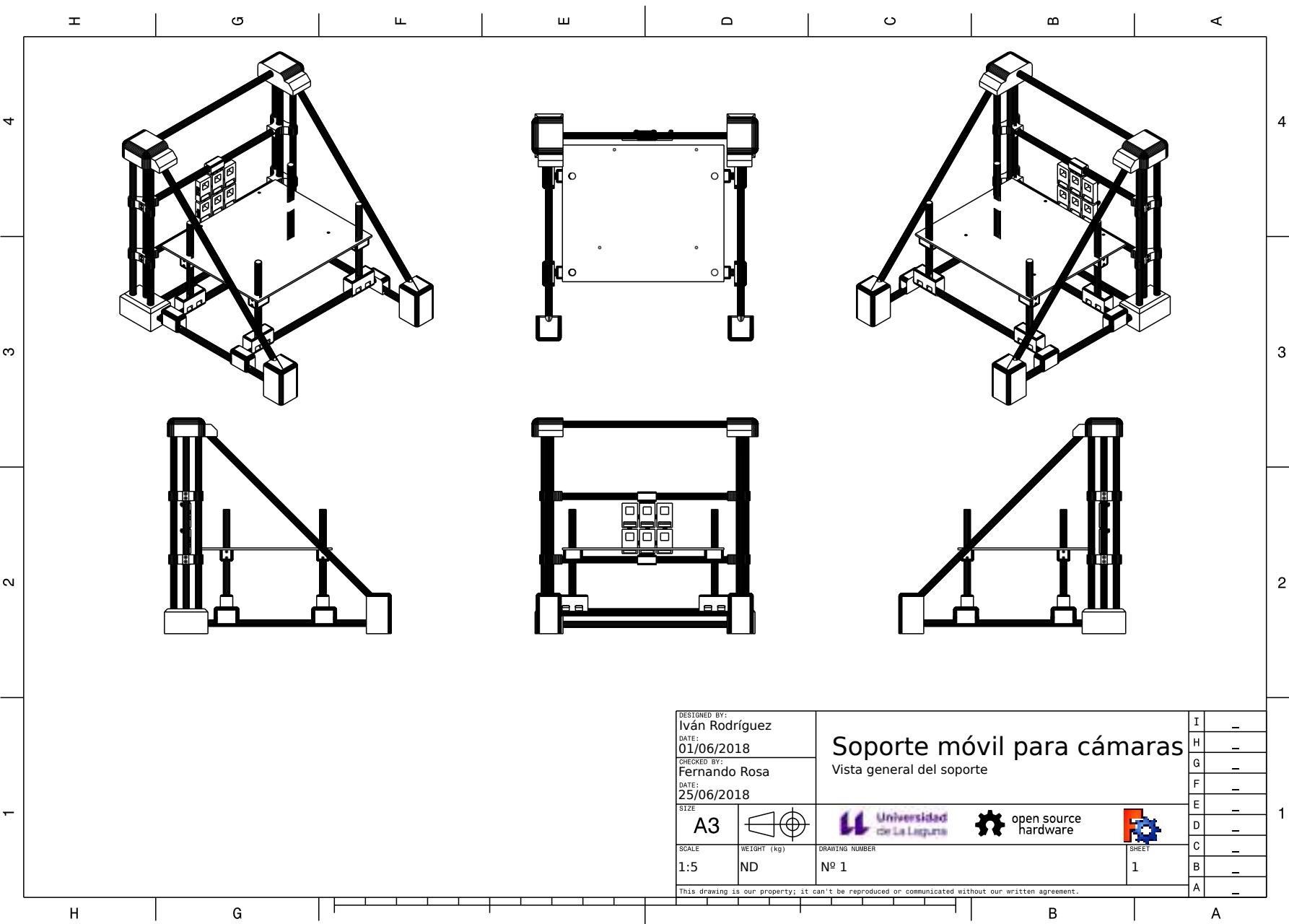
Tabla I.6: Características de los componentes del módulo de depuración de líneas.

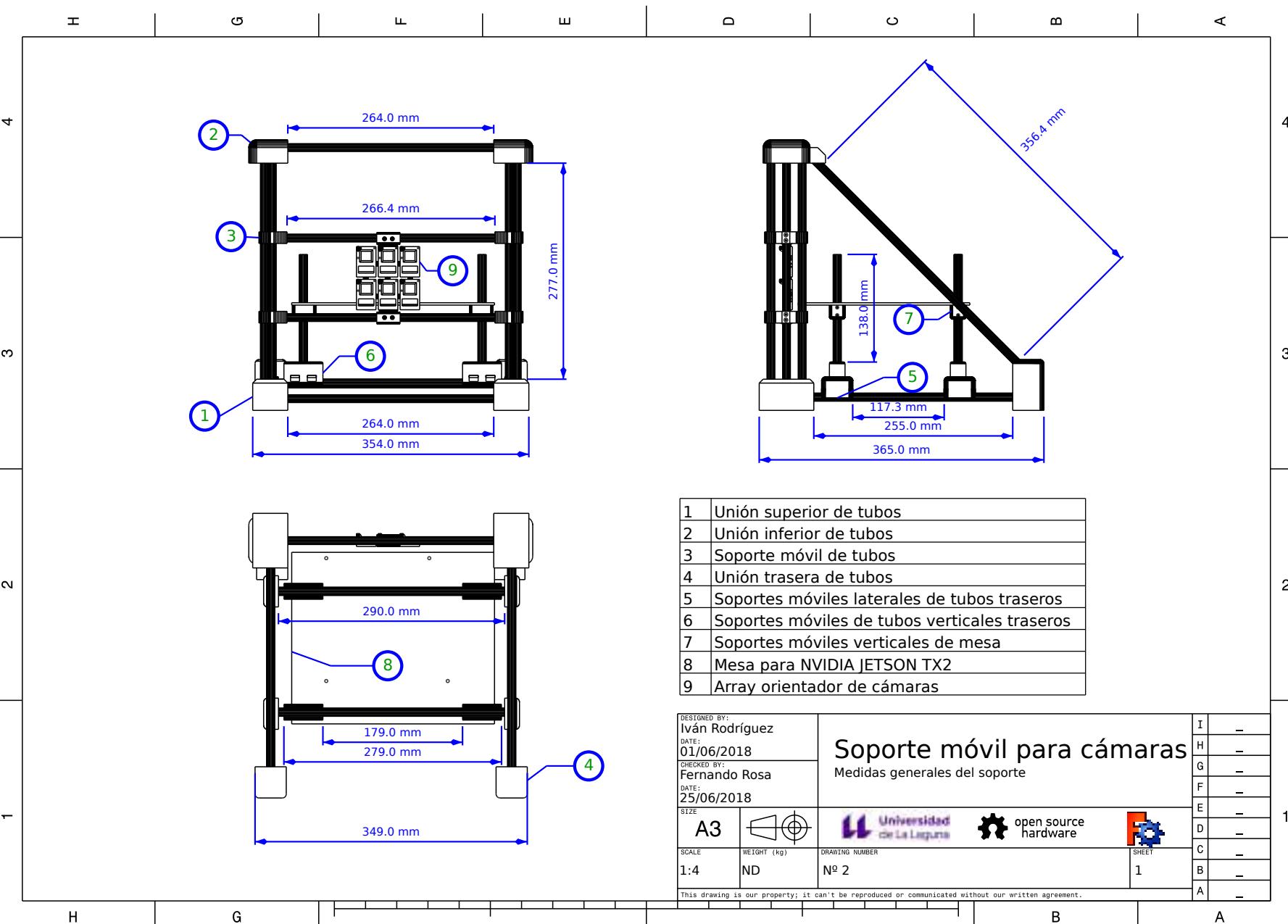
Anexo J

Planos de diseño del soporte móvil de cámaras

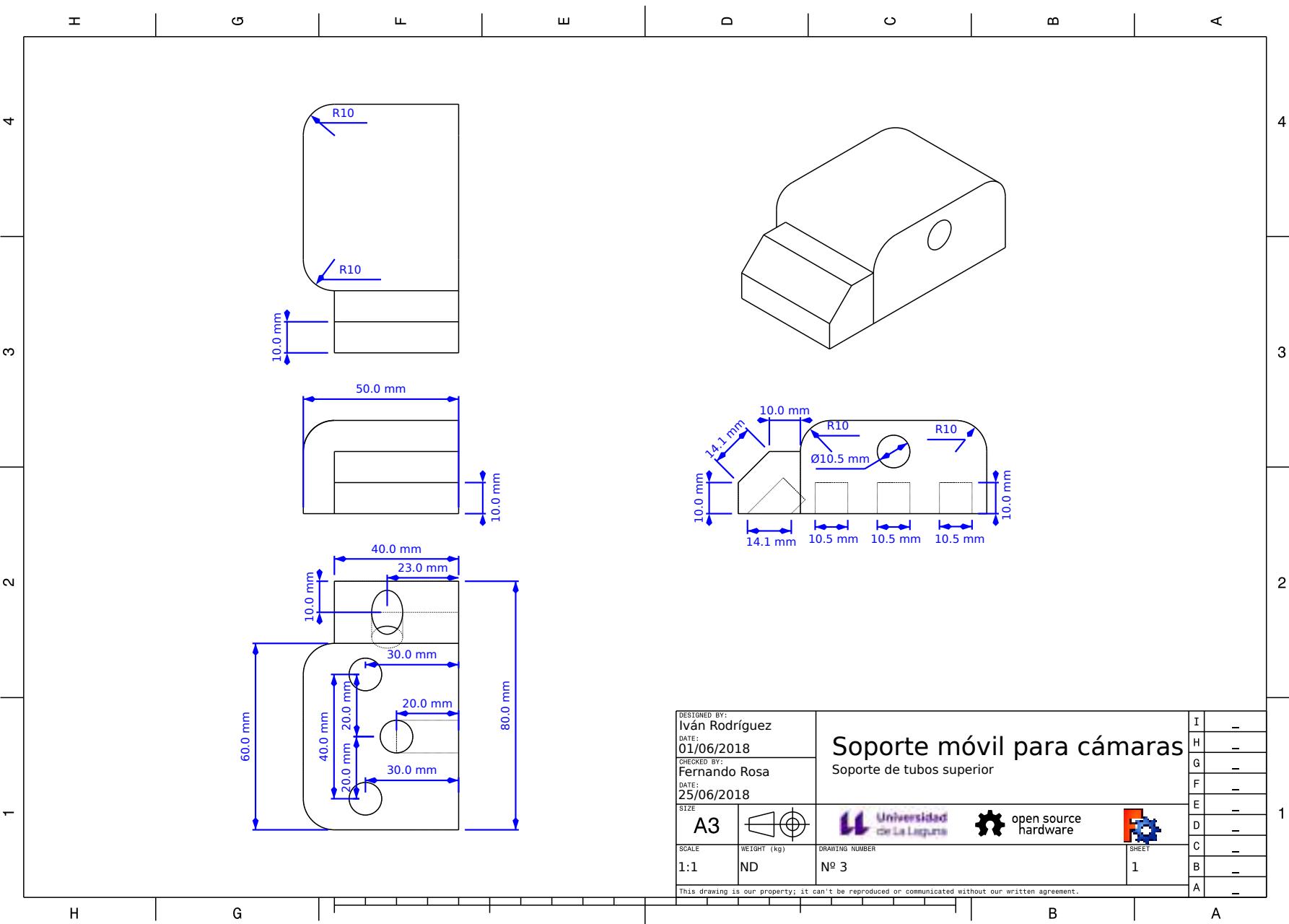
En este anexo, se presentan los planos de diseño del soporte móvil para la NVIDIA JETSON TX2 y las cámaras que incorpore.

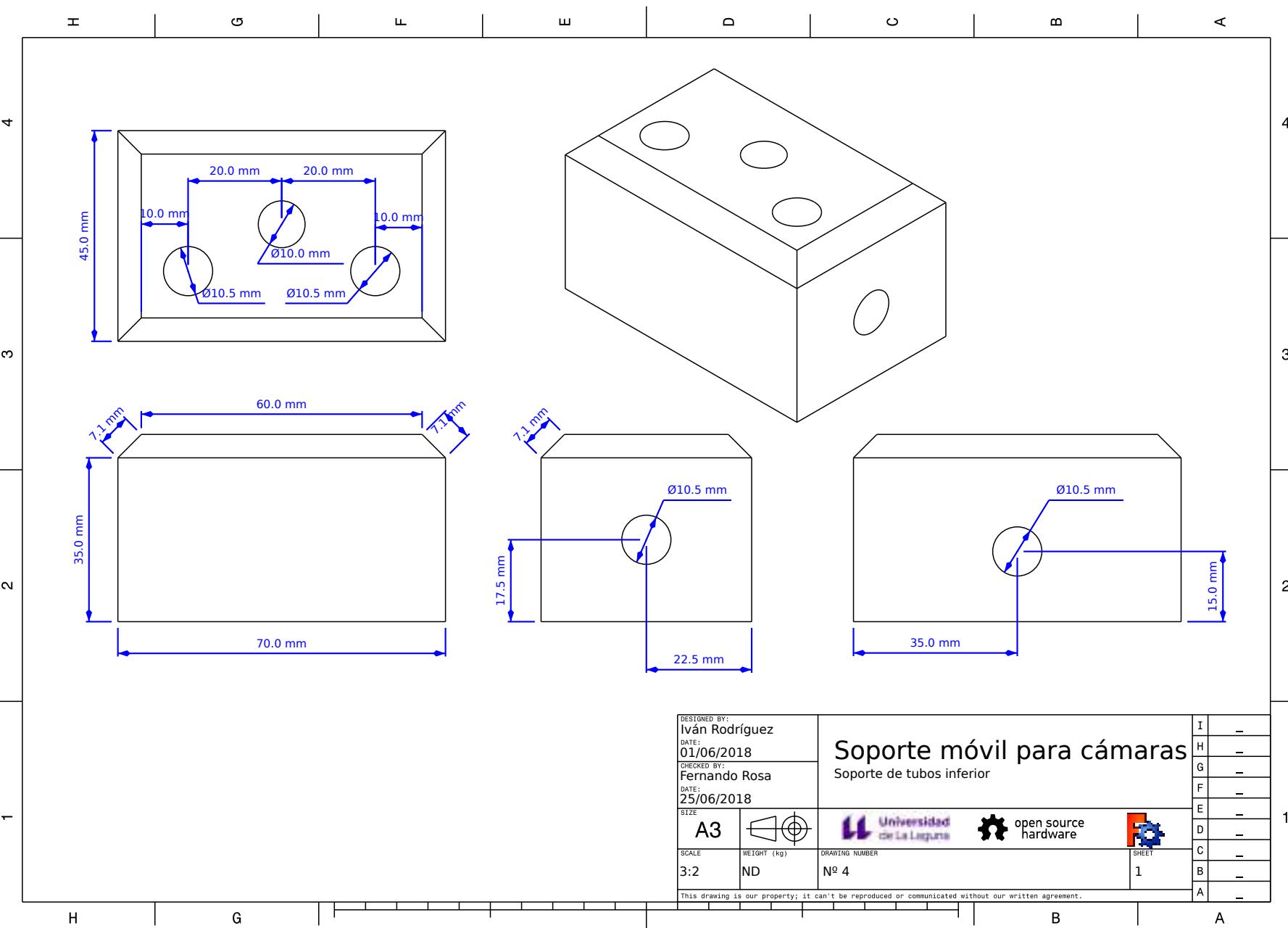
262 ANEXO J. PLANOS DE DISEÑO DEL SOPORTE MÓVIL DE CÁMARAS



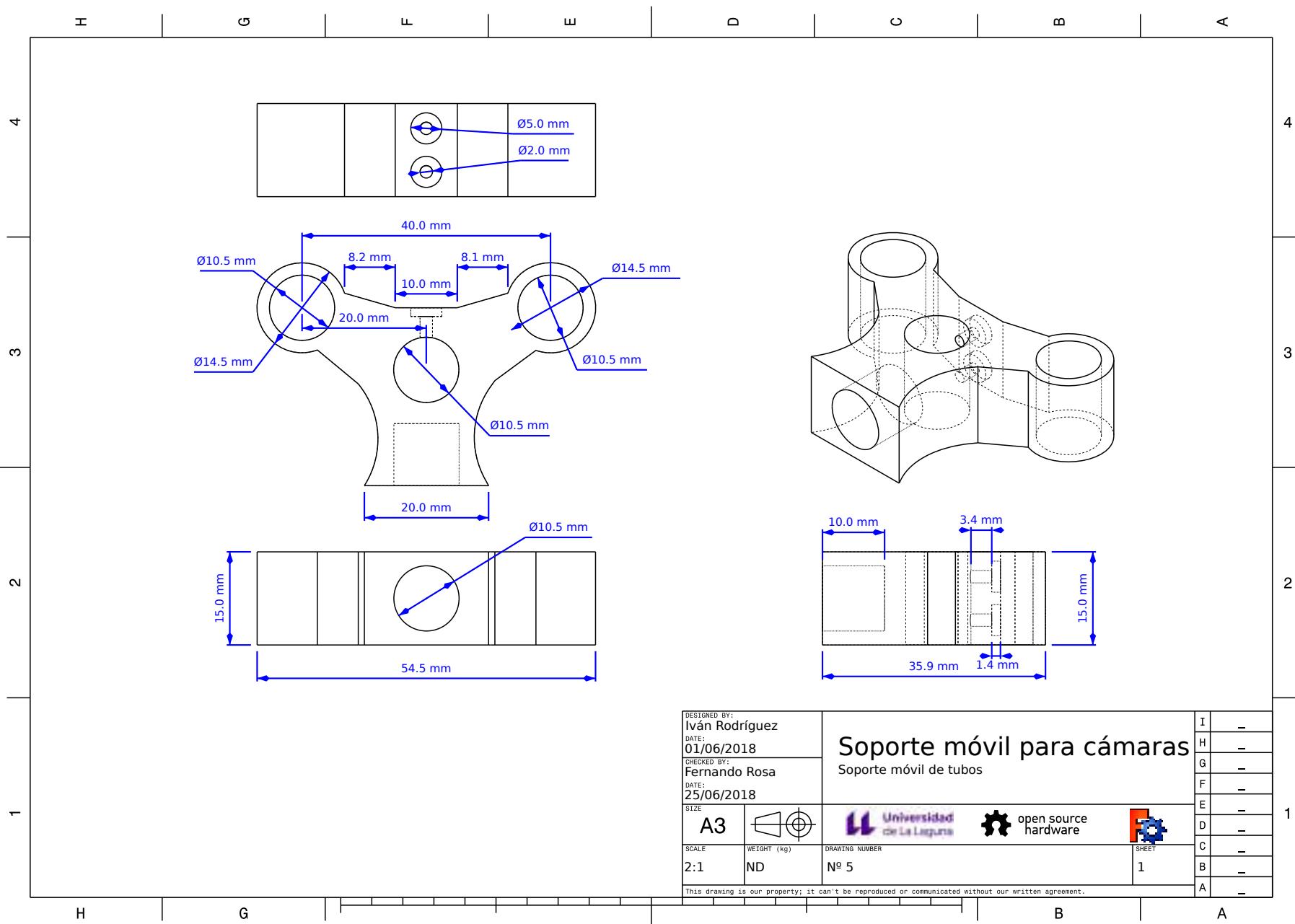


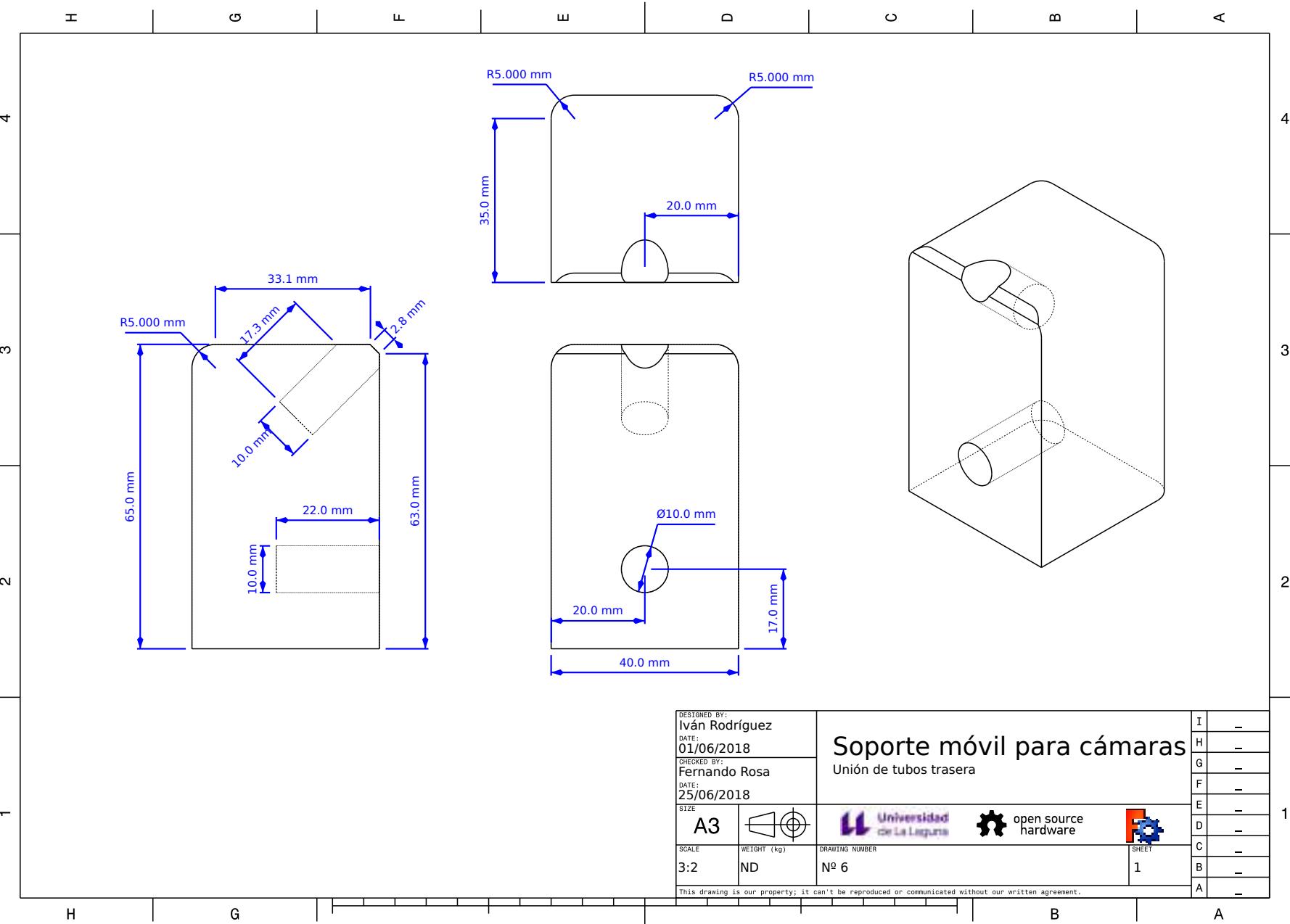
264 ANEXO J. PLANOS DE DISEÑO DEL SOPORTE MÓVIL DE CÁMARAS



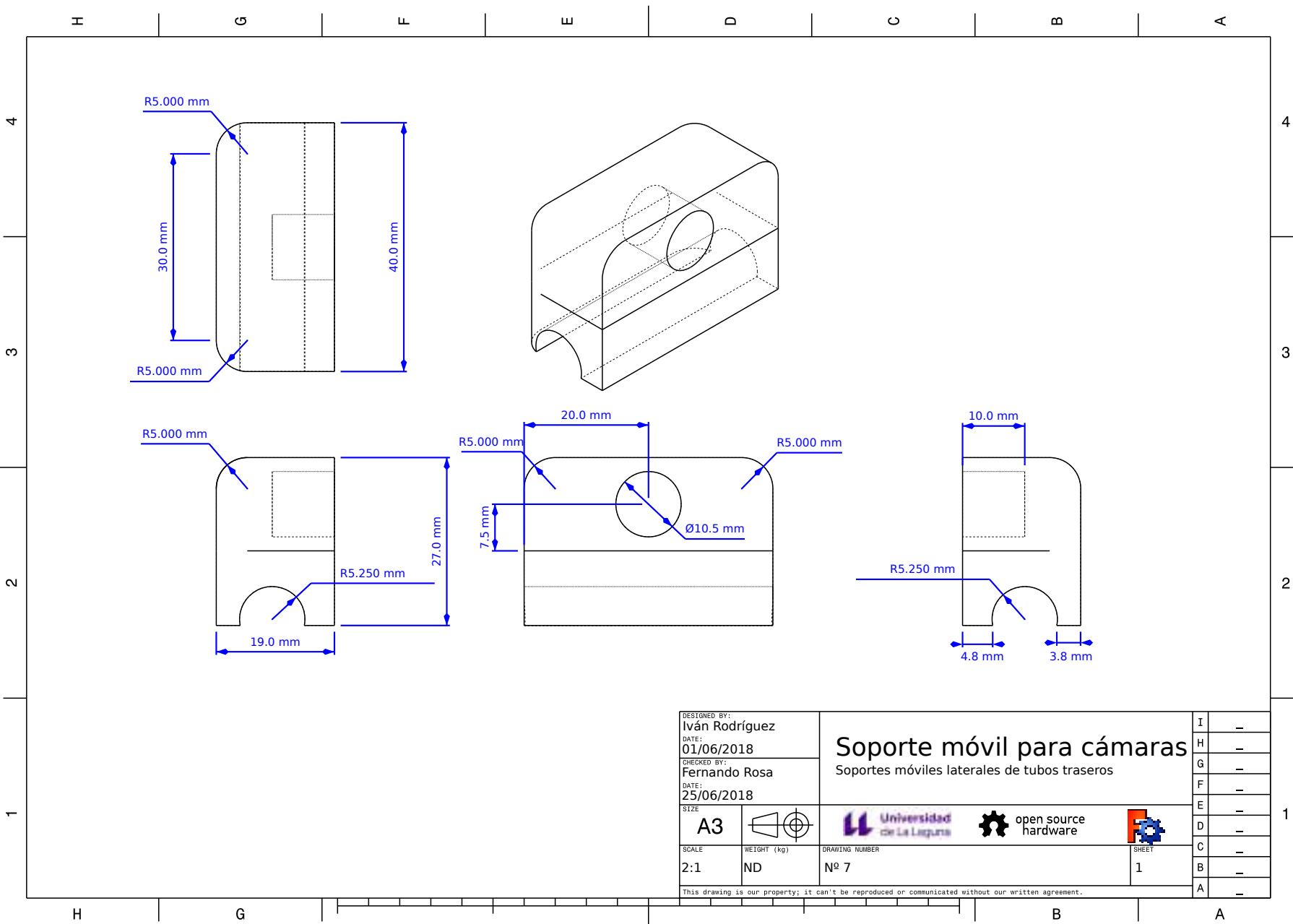


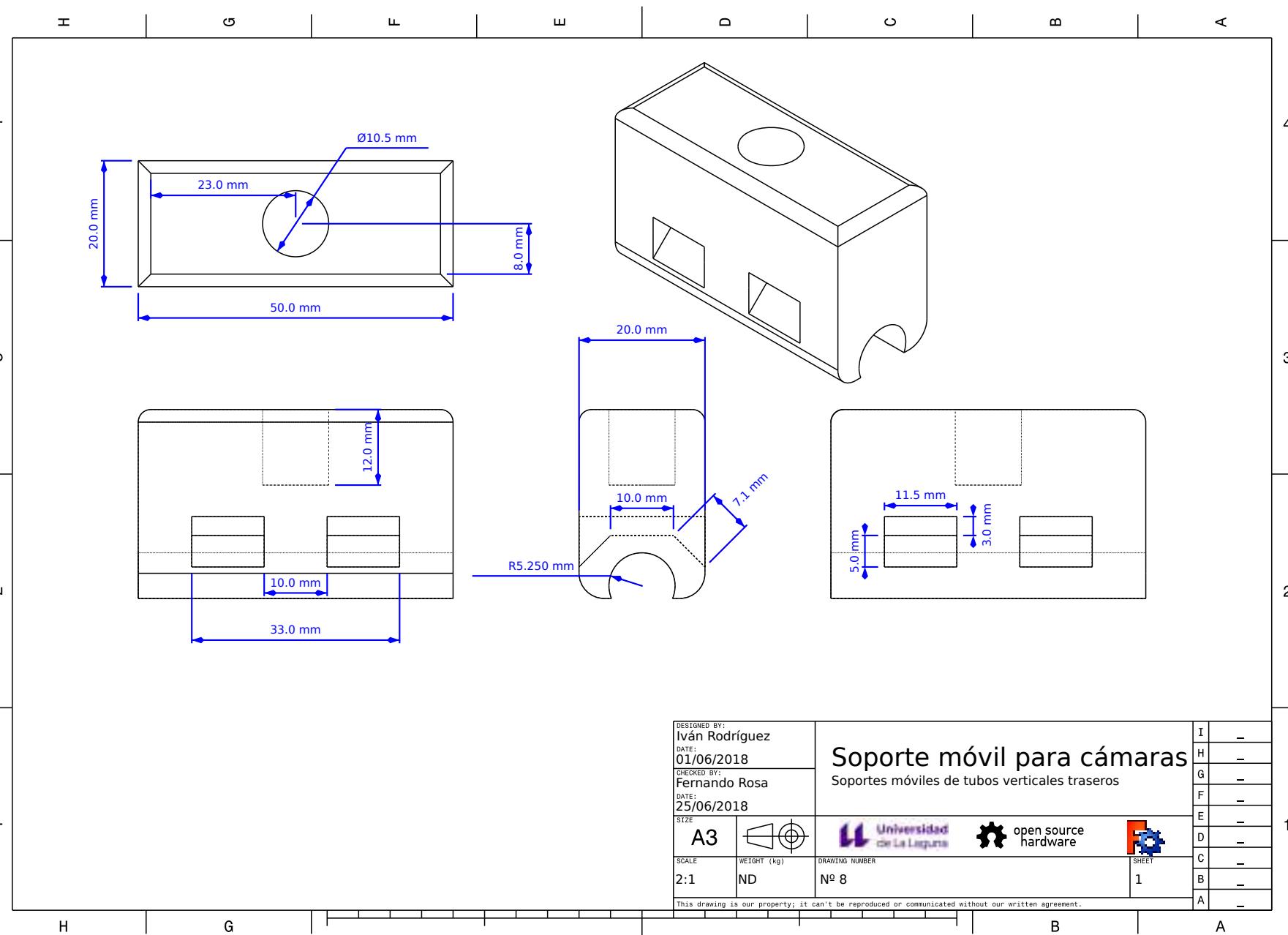
266 ANEXO J. PLANOS DE DISEÑO DEL SOPORTE MÓVIL DE CÁMARAS



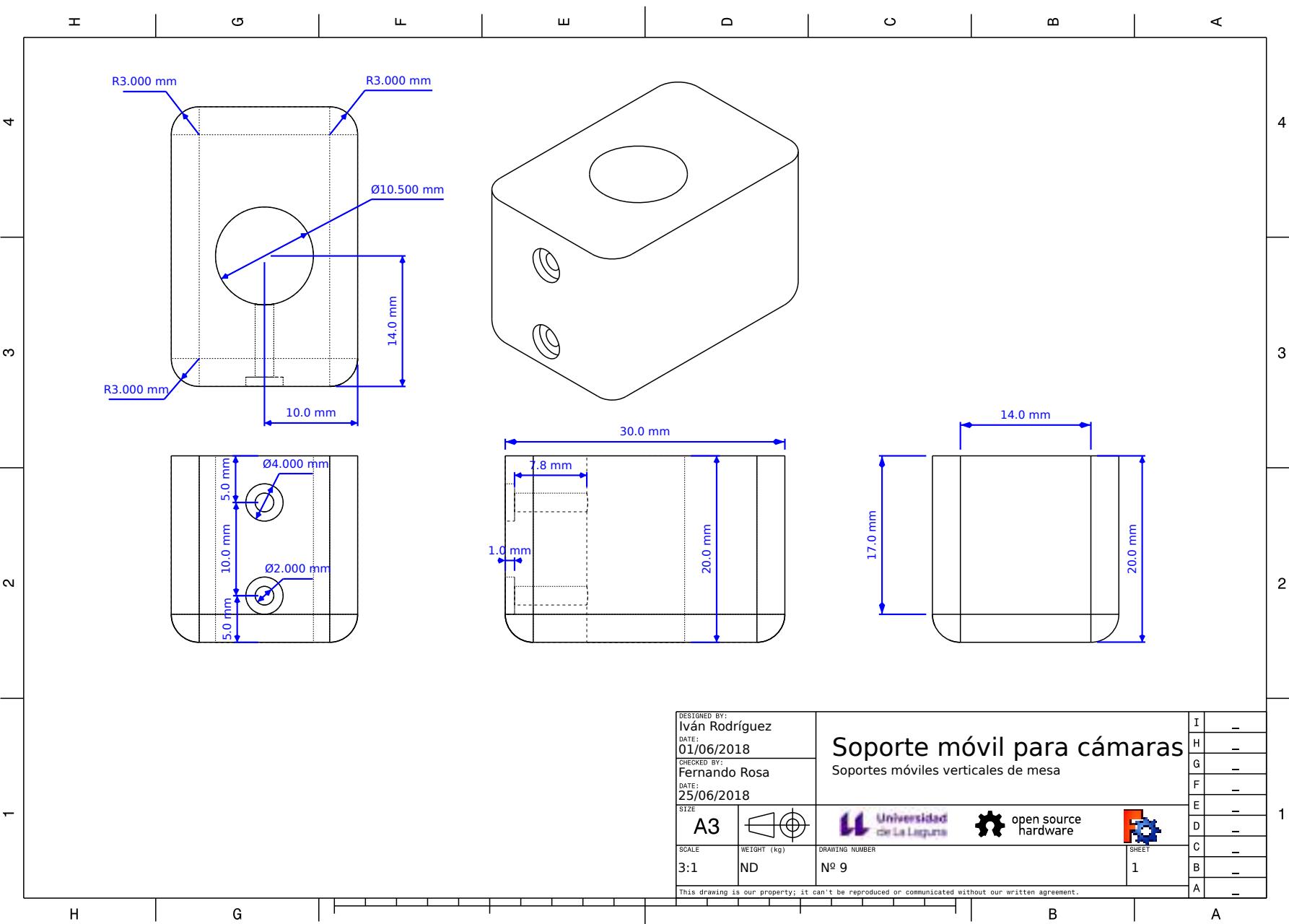


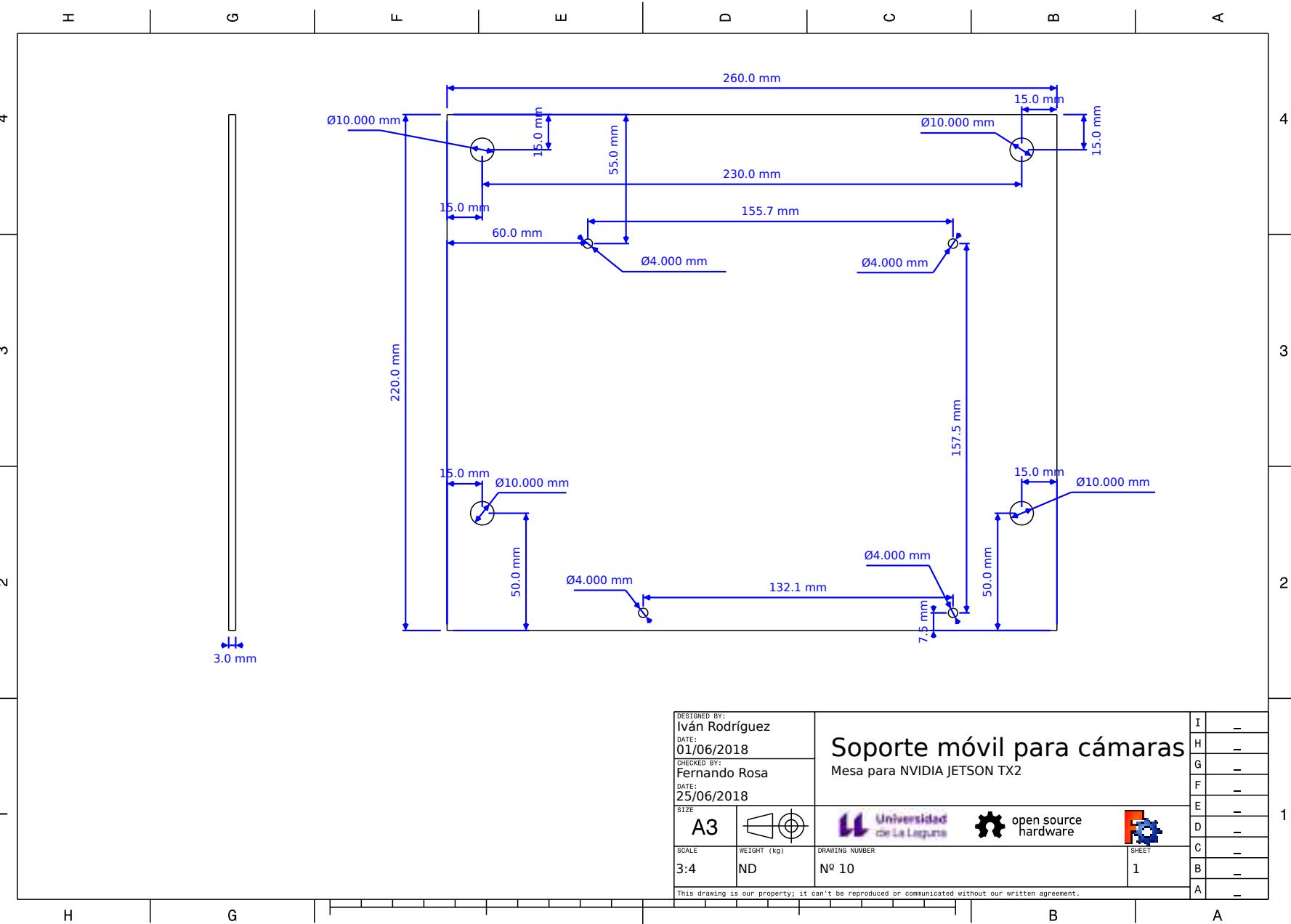
268 ANEXO J. PLANOS DE DISEÑO DEL SOPORTE MÓVIL DE CÁMARAS





270ANEXO J. PLANOS DE DISEÑO DEL SOPORTE MÓVIL DE CÁMARAS





272 ANEXO J. PLANOS DE DISEÑO DEL SOPORTE MÓVIL DE CÁMARAS

