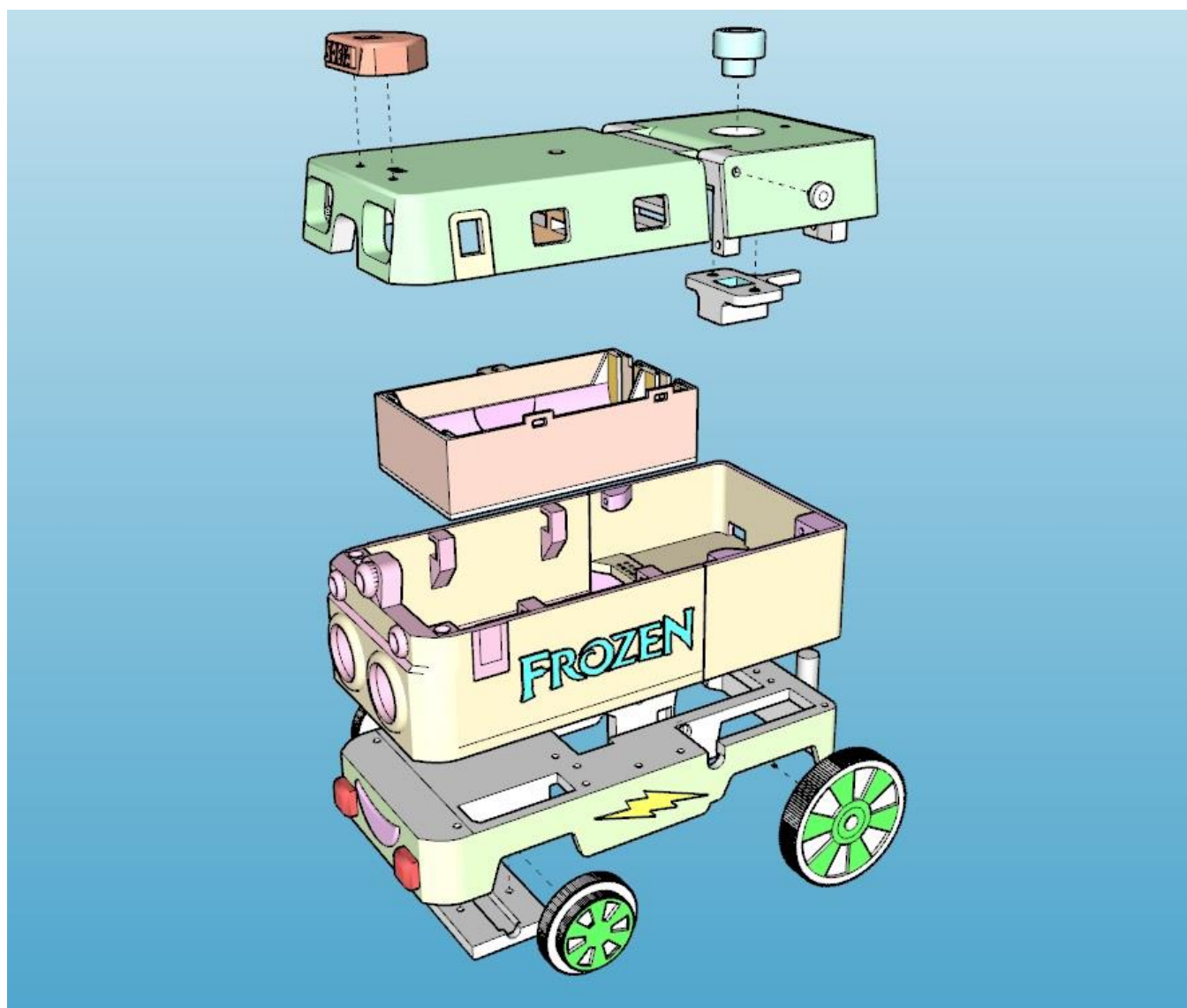


LOCOMOTORA “SOFÍA” COMPATIBLE CON TREN DE LEGO DUPLO

* DESCRIPCIÓN GENERAL:

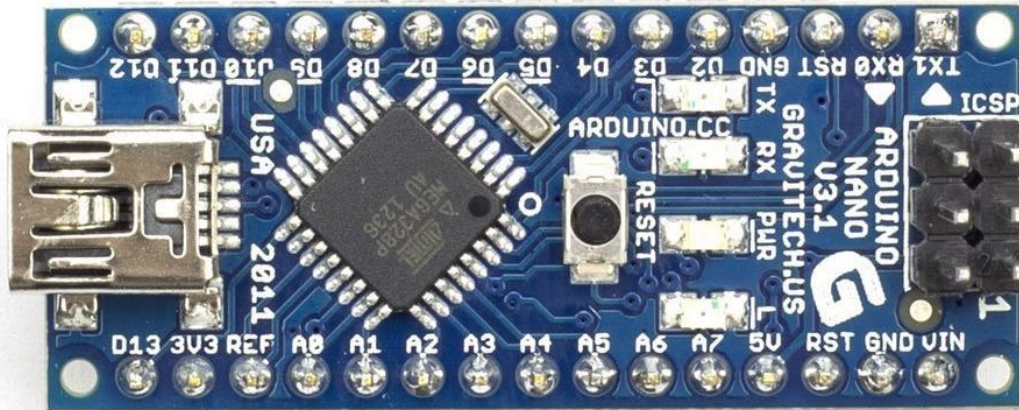
- La locomotora “Sofía” lleva un **Arduino Nano v3.0**, es controlado por un **mando por infrarrojos** y lleva un **sensor de distancia por ultrasonidos** el cual nos puede variar la velocidad y parar en caso de detectar un obstáculo.
- Lleva **2 luces Led’s** blancas delanteras para iluminación, **más una central** verde de señalización de "Control de velocidad por ultrasonidos Activo".
- Tiene **un motor para las ruedas motrices** (traseras) y un segundo **mini motor con reductora metálica** de bajas revoluciones (30 r.p.m) , el cual el eje va situado en la parte superior de la locomotora para propósitos variados. Yo le he situado una rueda en interior con imanes de neodimio para posibles juegos magnéticos.... Éstos motores están controlados por el driver **HG7881 (L9110S)**. También podemos bajar o subir la velocidad manualmente cuando el control de velocidad por ultrasonidos lo tenemos desactivado.
- Lleva **un interruptor de encendido** y un pequeño **altavoz activo** (reproduce pitidos) para señalar diversas opciones.
- Está alimentado por **2 baterías de Litio recargables tipo 18650**.
- Podéis encontrar las piezas para construirlo en: <https://www.thingiverse.com/thing:4012245>



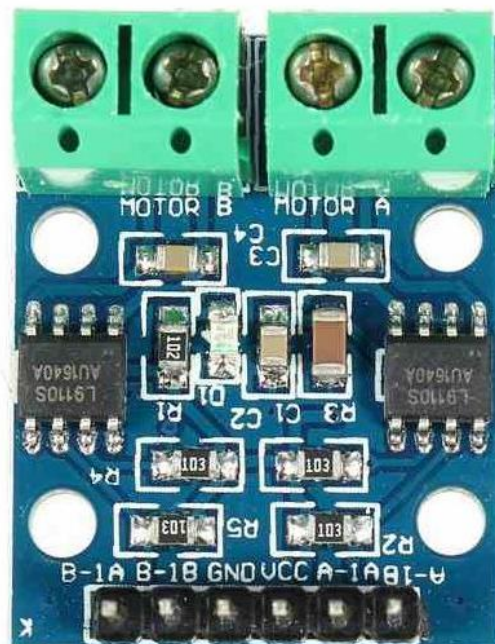
- **COMPONENTES PARA EL MONTAJE:**

- **NOTA: LOS ENLACES AQUÍ PUESTOS SON SIMPLEMENTE ORIENTATIVOS, TENDREIS QUE BUSCAR EN INTERNET (ALIEXPRESS, EBAY, AMAZON, ETC...), ESTOS COMPONENTES, QUE SEGURAMENTE LOS ENCONTRÉIS MÁS ECONÓMICOS.**

- **Un Arduino Nano v3.0** (Foto de v3.1) , podéis encontrarlo en muchos sitios, algo como esto:
<https://www.ebay.es/itm/NANO-V3-0-ATmega328P-CH340-SOLDADO-100-Compatible-con-Arduino-B0006/201786982926?ssPageName=STRK%3AMEBIDX%3AIT&trksid=p2060353.m2749.l2649>

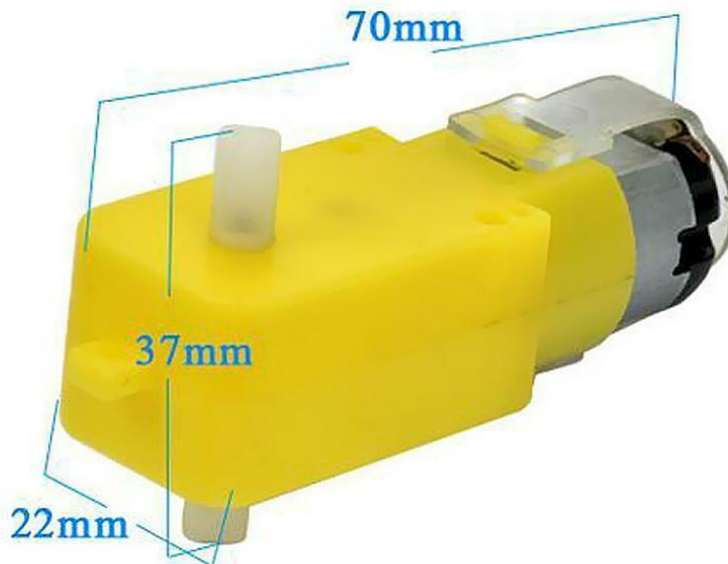


- **Un controlador o driver de 2 motores de C.C. modelo HG7881 (L9110S)**, algo como esto:
<https://www.ebay.es/itm/DC-Motor-Treiber-Board-Stepper-Treiber-HG7881-For-Arduino-Raspberry-Pi/112506068058?ssPageName=STRK%3AMEBIDX%3AIT&trksid=p2060353.m2749.l2649>



- **Un motor de corriente continua con eje doble** para montar las 2 ruedas motrices (traseras), de éste modelo (enlace para un pack de 10 motores) con Velocidad de giro: 1:48:

<https://www.ebay.es/itm/10-piezas-Intelligent-Car-TT-Motor-lanzamiento-Smart-Car-Gear-Motor-para/232758645073?ssPageName=STRK%3AMEBIDX%3AIT&trksid=p2060353.m2749.l2649>



- **Un segundo mini motor con reductora metálica de bajas revoluciones 30 rpm**, para la colocación de una rueda porta imanes en parte superior del modelo, algo como esto: (Nota: en éste enlace podemos escoger 2 modelos 1 de 6v y 30rpm (el usado en éste proyecto) y otro de 300rpm y 12v.

- <https://www.ebay.es/itm/Low-Gear-Motor-300-600-30RPM-Metal-Shaft-Robot-Boat-Wheel-Speed-Reduction-Mini/123884311603?ssPageName=STRK%3AMEBIDX%3AIT&var=424627833285&trksid=p2060353.m2749.l2649>



- **2 baterías de Litio recargables modelo 18650** para la alimentación de la locomotora algo como esto:
- <https://www.ebay.es/itm/1-2-4-6-10pcs-18650-Bater%C3%ADa-6800mAh-3-7V-Li-ion-Rechargeable-Battery-Flashlight/182754030072?ssPageName=STRK%3AMEBIDX%3AIT&var=690072577940&trksid=p2060353.m2749.12649>
- . NOTA: Estas baterías necesitan un cargador aparte para poder recargarlas, podréis encontrar algunos pack económicos de estas baterías que ya incluyen el cargador, por internet.



- Un Sensor de distancia por Ultrasonidos modelo **HC-SR04**, algo como esto:
- <https://www.ebay.es/itm/SENSOR-DE-DISTANCIA-ULTRASONIDO-HC-SR04-ARDUINO-ELECTRONICA-HCSR04-desde-Espa%C3%B1a/254287734886?ssPageName=STRK%3AMEBIDX%3AIT&trksid=p2060353.m2749.12649>



- Un kit de sensor de infrarrojos con mando a distancia (modelo VS1838) algo como esto:
- <https://www.ebay.es/itm/Kit-modulo-IR-con-mando-distancia-Infrarrojo-control-remoto-Arduino-VS1838/223098800406?ssPageName=STRK%3AMEBIDX%3AIT&trksid=p2060353.m2749.l2649>



- **1 varilla metálica de 3mm** de espesor para el eje de las ruedas delanteras (las traseras no necesitan ya que van montadas sobre el propio motor), las de éste ejemplo son de 140mm de largo, después tendremos que cortarla a la medida adecuada (**sobre 48mm de largo**), algo como esto:
- <https://www.ebay.es/itm/5pcs-varillas-redondas-de-transmisi%C3%B3n-3mm-acero-inoxidable-14cm-RC-Helic%C3%B3ptero/382195635336?ssPageName=STRK%3AMEBIDX%3AIT&trksid=p2060353.m2749.l2649>
- También necesitamos **una varilla de 2 mm** de espesor (éstas son de 160mm) tendremos que cortarlas a la distancia de **51mm** aproximadamente para hacer de bisagra de la tapa superior de la locomotora, algo como esto:
- <https://www.ebay.es/itm/10pcs-varillas-redondas-torno-de-transmision-hardware-acero-inoxidable-160x2mm/153288265370?hash=item23b0b14e9a:g:DhoAAOSwpoxdOLrW>



- **2 diodos led blancos** (3,6v cada uno) para la iluminación delantera, y **un diodo led verde** (2,1V) para la señalización de “control por ultrasonidos” activo. Ambos son de 5mm de grosor, más 2 resistencias, **una de 47 ohms** para las dos luces blancas y **otra de 150 ohms** para el led Verde de 1/4w.



- **Un altavoz activo de 12mm** de grosor y 5v algo como esto:
- <https://www.ebay.es/itm/5PCS-5V-12MM-Active-Buzzer-Magnetic-Long-Continuous-Beep-Tone-Alarm-Ringer-NEW/113224614462?ssPageName=STRK%3AMEBIDX%3AIT&trksid=p2060353.m2749.l2649>



- **Un interruptor (pulsador)** de encendido de 8mmx 8mm, algo como esto:
- <https://www.ebay.es/itm/lote-5-pcs-Interruptor-pulsador-cuadrado-8x8mm-On-Off-Arduino-prototipos/121481984872?hash=item1c48e3fb68:g:130AOSw54xUXdLC>



- **Tornillería métrica 2mm**, estos son de 10mm de largo, en alguna ocasión en el montaje es necesario cortarlos un poco, algo cómo esto:
- https://www.ebay.es/itm/HOTsell-100-pcs-M2-x-10mm-Screw-self-tapping-screw-Cross-Head-Screw-screws-Black/182169134472?_trkparms=aid%3D111001%26algo%3DREC.SEED%26ao%3D1%26asc%3D20140423084956%26meid%3D5cbfa91cee65481c9a02b3e477d43daf%26pid%3D100033%26rk%3D6%26rkt%3D8%26sd%3D323745148489%26itm%3D182169134472%26pmt%3D0%26noa%3D1%26pg%3D2045573&_trksid=p2045573.c100033.m2042



- **4 Imanes de Neodimio para el cierre magnético** de la tapa superior de tamaño 4x4mm cilíndricos, algo como ésto:
- https://www.ebay.es/itm/Imanes-Potentes-4x4mm-Neodimio-Cilindro-Disco-Manualidades-Iman-4mm-Diam-X-4mm/333395867768?hash=item4d9ff13878:m:miExx5AzaDoBQgyjZT2h_wQ
- **Nota:** Tener especial atención al montarlos para que queden las dos partes (tapa superior) y parte inferior con los polos opuestos (es decir, de modo que se atraigan). Si no la tapa nunca quedará cerrada.



- **6 imanes de Neodimio de 10x3mm** para el disco imantado situado en parte superior (para juegos magnéticos, opcional), algo como esto:
- https://www.amazon.es/Fridge-Magnets-Storage-Magnetic-Picture/dp/B07DNBxBWQ/ref=sr_1_15?_mk_es_ES=%C3%85M%C3%85%C5%BD%C3%95%C3%91&keywords=imanes+neodimio+4x4&qid=1575289278&s=kitchen&sr=1-15



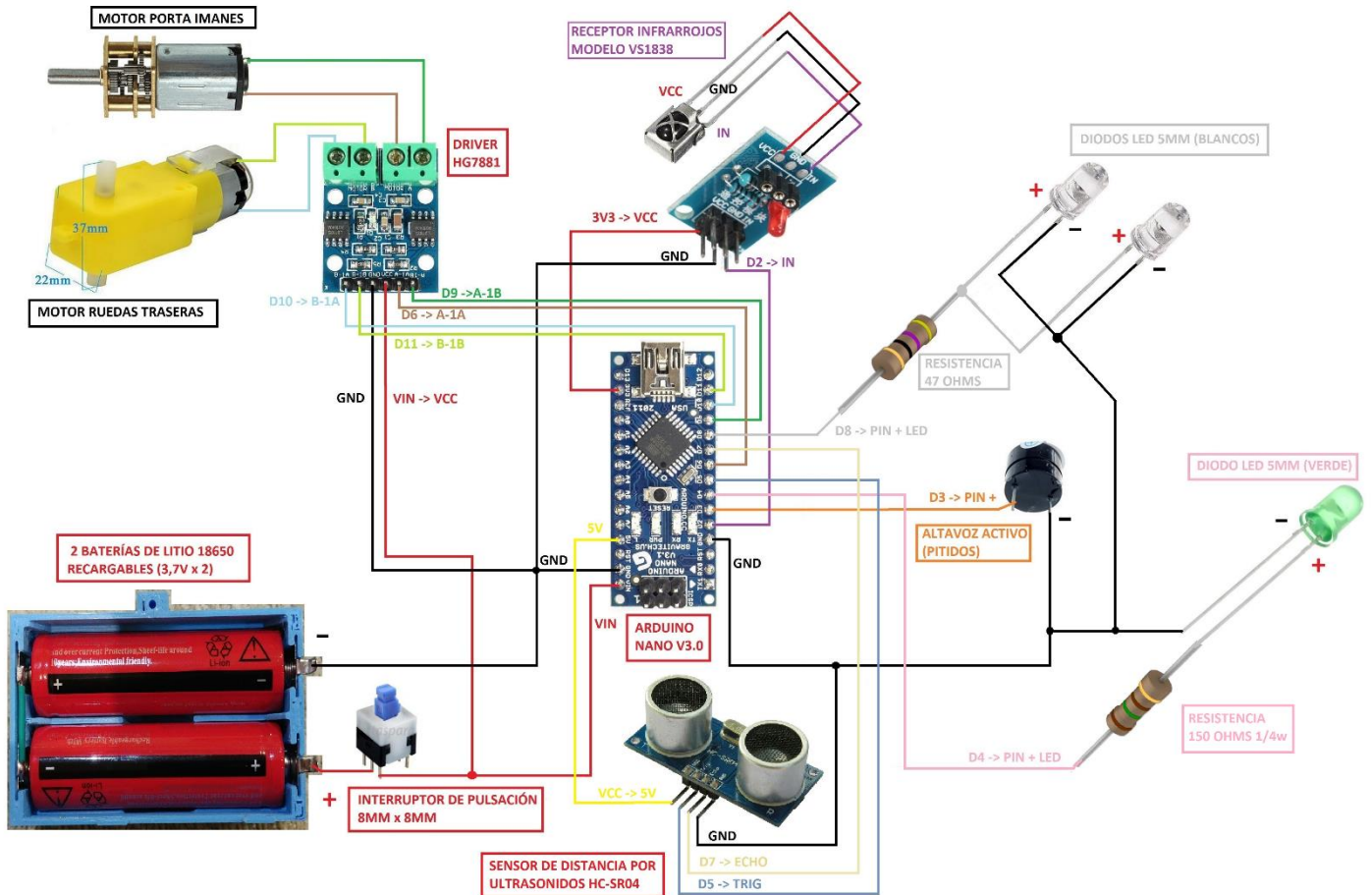
- **Cables tipo Dupont (Hembra–Hembra)** a poder ser de 2 longitudes (20cm y 10 cm), con esto realizamos todo el conexionado del proyecto y evitamos tener que realizar soldaduras (aún así es necesario realizar alguna). (Nota: Si encontramos de 15cm de longitud también nos viene bien para dejarlos sin mucho sobrante).
- <https://www.ebay.es/itm/40x-Cables-Hembra-Hembra-20cm-Jumpers-Dupont-2-54-Arduino-Protoboard-Prototipos/292549309025?hash=item441d4c3261:g:7NgAAOSwVYtb4U5a>
- https://www.ebay.es/itm/40-CABLES-HEMBRA-HEMBRA-10cm-jumpers-dupont-2-54-arduino-pic-protoboards/254290265041?_trkparms=aid%3D555018%26algo%3DPL.SIM%26ao%3D1%26asc%3D20131003132420%26meid%3D23bfafe2aecf4a3683c77f320f442c75%26pid%3D100005%26rk%3D10%26rkt%3D12%26mehot%3Dsb%26sd%3D202066717194%26itm%3D254290265041%26pmt%3D1%26noa%3D0%26pg%3D2047675&_trksid=p2047675.c100005.m1851



- Terminales para la batería (2 con muelle y 2 planos) de 12mm, algo cómo esto:
- https://www.amazon.es/sourcing-map-Terminal-Laminado-Contacto/dp/B07DC71JTC/ref=pd_sbs_107_4/257-8031535-2046002?encoding=UTF8&pd_rd_i=B07DC71JTC&pd_rd_r=7f801466-9f56-46bd-8281-118005eaa2ad&pd_rd_w=HVIOOr&pd_rd_wg=C8DDc&pf_rd_p=71620895-af79-4429-92c2-a38434153b95&pf_rd_r=XY3DKXPGSBH5N784WCA4&psc=1&refRID=XY3DKXPGSBH5N784WCA4



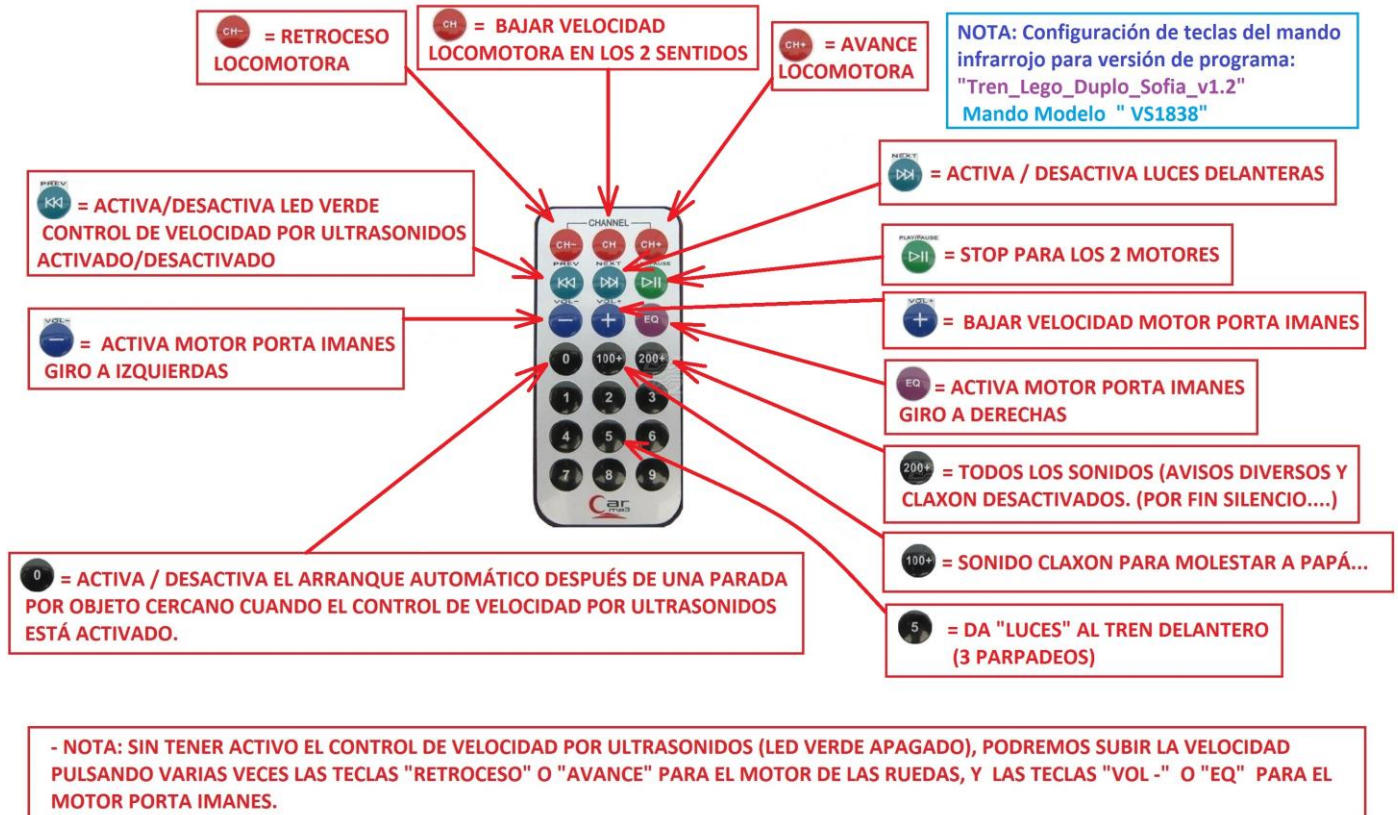
- ESQUEMA DE MONTAJE:**



- **Nota:** En caso de que al realizar el montaje y pulsemos la tecla del mando a distancia “CH+” y el motor de las ruedas nos gire hacia atrás, simplemente intercambiamos los dos cables que van al motor (podemos hacerlo en el bornero del driver **HG7881**).

• FUNCIONES PROGRAMADAS DEL MANDO A DISTANCIA POR INFRARROJOS:

- Si compramos el mismo modelo de mando infrarrojos (**modelo VS1838**) seguramente no tengamos que tocar nada a la programación, ya que por lo que he probado (3 modelos iguales) el código de emisión de cada tecla es idéntico. Entonces ésta sería la configuración para cada tecla programada (Configuración del mando para primeras versiones del programa "**Tren_Lego_Duplo_Sofia_v1.2**")



- En caso de no tener el mismo modelo de receptor infrarrojo, o tenemos otro mando diferente (**hasta podríamos utilizar el mando a distancia de un televisor**), ya que la librería de Arduino admite varias marcas conocidas, tendríamos que tocar en el programa que cargamos a nuestro Arduino Nano, para configurar cada tecla de nuestro propio mando para que funcione correctamente, explicaremos esto más adelante.

- Se ha añadido una última actualización, para que **podamos con un mando infrarrojo poder controlar 2 locomotoras** (evidentemente tendremos que construir 2), para ello **tenemos que cargar a cada locomotora un programa diferente.**

- El programa para la **Locomotora 1** es -> **“Tren_Lego_Duplo_Sofia_v1.2_Tren1”**
- El programa para la **Locomotora 2** es -> **“Tren_Lego_Duplo_Sofia_v1.2_Tren2”**
- Esta sería la configuración de cada tecla para trabajar con las 2 locomotoras:

CONFIGURACION DEL MANDO INFRARROJO (MODELO VS1838) PARA EL CONTROL DE 2 LOCOMOTORAS. ES NECESARIO CARGAR UN PROGRAMA DIFERENTE A CADA LOCOMOTORA
PROGRAMA LOCOMOTORA 1 --> **Tren_Lego_Duplo_Sofia_v1.2_TREN1.ino** PROGRAMA LOCOMOTORA 2 --> **Tren_Lego_Duplo_Sofia_v1.2_TREN2.ino**

CH- = RETROCESO LOCOMOTORA 1
(* CON VARIAS PULSACIONES SUBE DE VELOCIDAD)

CH = BAJA LA VELOCIDAD EN AVANCE Y RETROCESO DE LOCOMOTORA 1 (*) MEDIANTE PULSACIONES

CH+ = AVANCE LOCOMOTORA 1
(* CON VARIAS PULSACIONES SUBE LA VELOCIDAD)

KK = ACTIVA O DESACTIVA "LED VERDE" EN LOCOMOTORA 1 CONTROL DE VELOCIDAD POR ULTRASONIDOS
ACTIVADO O DESACTIVADO

Next = ACTIVA O DESACTIVA LUCES DELANTERAS EN LOCOMOTORA 1 Y 2

Stop = STOP PARA LOS 2 MOTORES DE LOCOMOTORA 1

0 = MOTOR PORTA IMANES 1 GIRO A IZQUIERDAS (CON VARIAS PULSACIONES SUBE DE VELOCIDAD)

+ = BAJA LA VELOCIDAD MOTOR PORTA IMANES 1 MEDIANTE PULSACIONES

EQ = MOTOR PORTA IMANES 1 GIRO A DERECHAS (CON VARIAS PULSACIONES SUBE DE VELOCIDAD)

0 = ACTIVA O DESACTIVA EL ARRANQUE AUTOMÁTICO DESPUÉS DE UNA PARADA POR OBJETO CERCAÑO CUANDO EL CONTROL DE VELOCIDAD POR ULTRASONIDOS ESTÁ ACTIVADO EN LOCOMOTORA 1 Y 2

100+ = SONIDO DE "CLAXON" PARA MOLESTAR A PAPÁ DE LAS LOCOMOTORAS 1 Y 2 (3 PITIDOS LARGOS)

200+ = TODOS LOS SONIDOS DESACTIVADOS EN LOCOMOTORA 1 Y 2 (AVISOS DIVERSOS Y CLAXON)



1 = RETROCESO LOCOMOTORA 2
(* CON VARIAS PULSACIONES SUBE DE VELOCIDAD)

2 = BAJA LA VELOCIDAD EN AVANCE Y RETROCESO DE LOCOMOTORA 2 (*) MEDIANTE PULSACIONES

3 = AVANCE LOCOMOTORA 2
(* CON VARIAS PULSACIONES SUBE LA VELOCIDAD)

4 = ACTIVA O DESACTIVA "LED VERDE" EN LOCOMOTORA 2 CONTROL DE VELOCIDAD POR ULTRASONIDOS
ACTIVADO O DESACTIVADO

5 = DA "LUCES" 3 PARPADEOS DE LUCES DELANTERAS EN LOCOMOTORA 1 Y 2 (PARA AVISAR QUE ESTÁ LA GUARDIA CIVIL MULTANDO..... JE,JE)

6 = STOP PARA LOS 2 MOTORES DE LOCOMOTORA 2

7 = MOTOR PORTA IMANES 2 GIRO A IZQUIERDAS (CON VARIAS PULSACIONES SUBE DE VELOCIDAD)

8 = BAJA LA VELOCIDAD MOTOR PORTA IMANES 2 MEDIANTE PULSACIONES

9 = MOTOR PORTA IMANES 2 GIRO A DERECHAS (CON VARIAS PULSACIONES SUBE DE VELOCIDAD)

(*) --> PARA PODER SUBIR O BAJAR LA VELOCIDAD DE LAS LOCOMOTORAS MANUALMENTE (MEDIANTE PULSACIONES) "SOLO PARA EL MOTOR DE LAS RUEDAS". TENEMOS QUE TENER EL CONTROL DE VELOCIDAD POR ULTRASONIDOS DESACTIVADO (LED VERDE APAGADO, TECLAS **KK** Y **4** PARA DESACTIVARLO).

- Se utilizan todas las teclas del mando infrarrojo, **algunas trabajan para las 2 locomotoras al mismo tiempo** debido a que no se dispone de suficientes teclas para poder individualizar todas las funciones. **Estas son:**

- La tecla número **“5”** **5** (Dar “Luces” 3 Parpadeos de luces delanteras)

- La tecla **“Next”** **Next** (Activa o desactiva las Luces Delanteras)

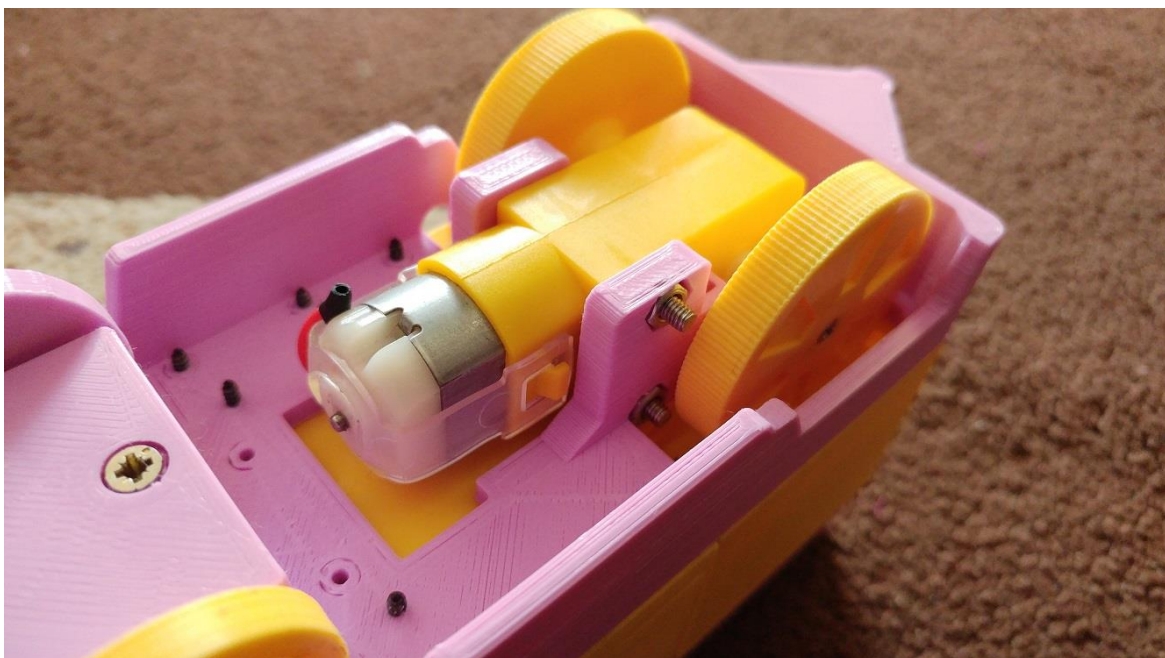
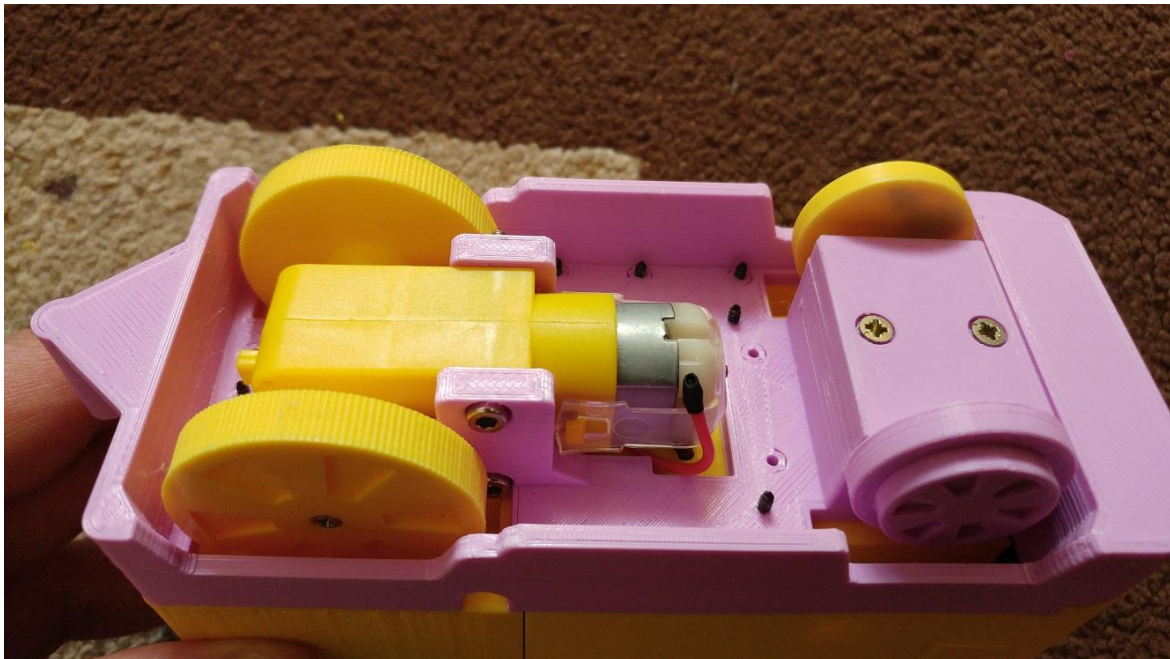
- La tecla **“0”** **0** (Activa o desactiva el arranque automático después de una parada por objeto cercano cuando el control de velocidad por ultrasonidos está activado).

- La tecla **“100+”** **100+** (Sonido de “Claxon” 3 pitidos largos).

- La tecla **“200+”** **200+** (Todos los sonidos desactivados)

- **MONTAJE (CON ALGUNAS FOTOGRAFÍAS):**

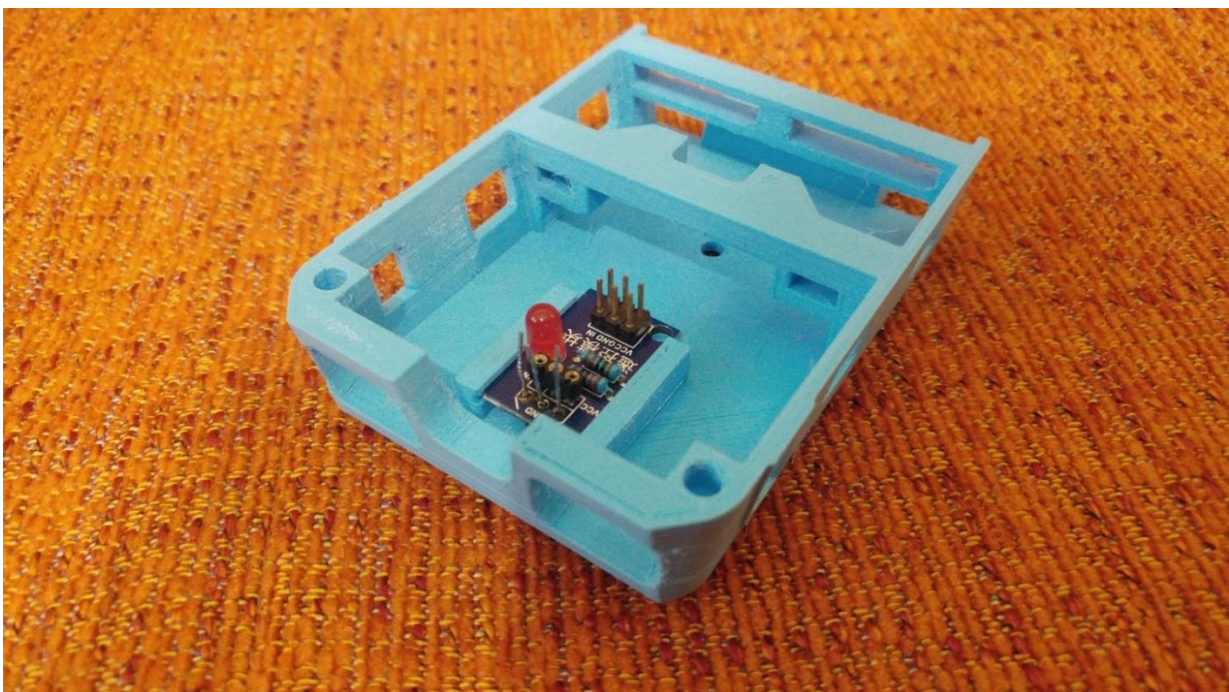
- **Nota:** Los diseños de las piezas que se pueden ver en alguna de éstas fotografías pueden ser algo diferentes a las piezas finales dejadas en Thingiverse, ya que se han hecho pequeñas modificaciones desde el diseño inicial.
- Primeramente **acoplamos las ruedas traseras** (motrices) al eje de la reductora del motor y las fijamos con un tornillo tipo rosca madera, creo recordar de 2mm. La fijación del motor al chasis de la locomotora es con tornillería de M3x30 con sus respectivas tuercas, (entran algo justos debido al tamaño de la rueda trasera). **Nota: Debemos tener ya soldados los dos cables que van al motor, dejando una longitud suficiente para que lleguen hasta el driver HG7881 (L9110S), 15cm es más que suficiente.**
- **Para el montaje de las ruedas delanteras** necesitamos cortar una varilla de 3mm de grosor a aproximadamente 48mm de largo. Las ruedas delanteras entran a presión en ésta varilla, aunque podemos darle algo de pegamento tipo Loctite para fijarlas más robustamente. Posteriormente montaremos la tapa que evita que se salga la varilla con tornillería de 2mm tipo rosca madera.



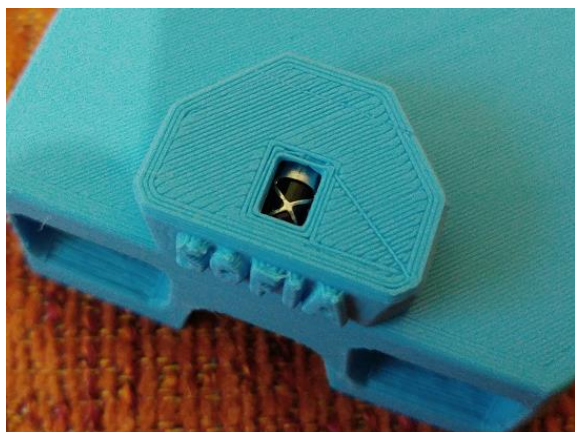
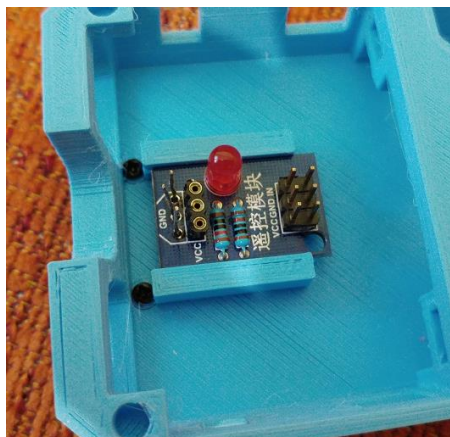
- Para la parte superior de la locomotora, aquí es donde va montado el receptor del mando a distancia por infrarrojos y el motor porta imanes de neodimio, **tenemos que tener especial atención a la posición en la que colocamos el diodo receptor de infrarrojo**. Debe ir cómo se ve en la fotografía, las patillas hacia el centro de la pieza y el diodo tumbado hacia la parte delantera de la pieza.



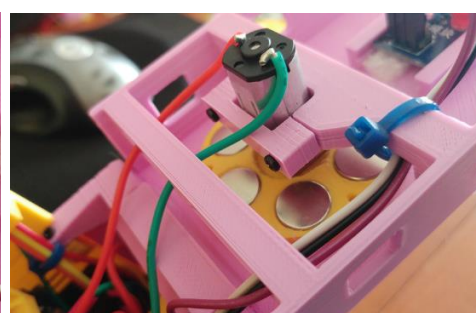
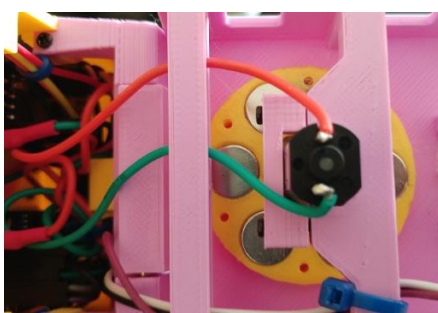
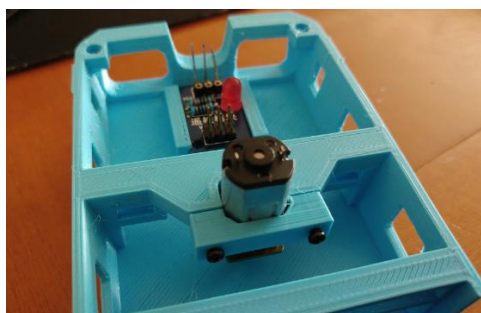
- En la parte interior colocamos la plaquita que viene en el kit del mando a distancia con infrarrojos y con algo de pericia introducimos las patillas del diodo infrarrojo por los huecos en la placa, posteriormente **tendremos que soldarlas para dejarlas fijadas**.
- En las patillas de conexión que tenemos en la placa usaremos los cables tipos Dupont Hembra-Hembra que irán hasta nuestro Arduino Nano, llevándolos por los huecos laterales de la pieza, para que no interfieran con el movimiento del motor porta imanes. **Tener en cuenta que la alimentación (VCC) del sensor infrarrojo va a la patailla de 3V3 de nuestro Arduino Nano**.
- Ésta pieza lleva 2 imanes de neodimio de 4mmx4mm cilíndricos en los huecos delanteros para el cierre magnético con la parte inferior y va sujeta mediante una varilla de 2mm de espesor que hace de bisagra a la parte superior trasera de aproximadamente 51mm de longitud.



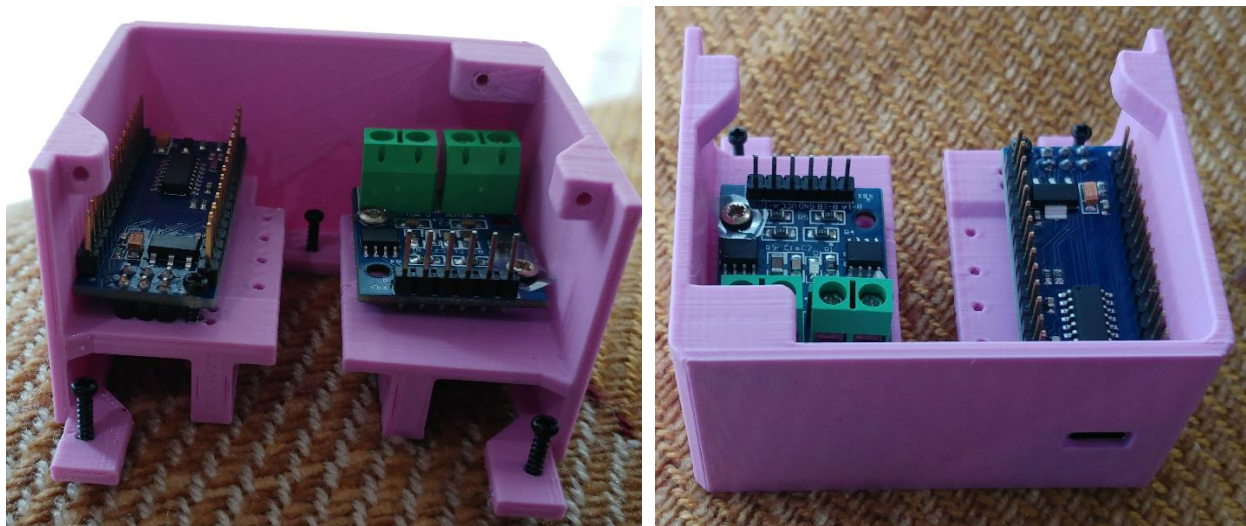
- Para el montaje de la protección "SOFIA" del diodo infrarrojo en parte superior necesitamos 2 tornillos de 2mm algo cortos para no sobresalir (yo he utilizado los que salen en la descripción del material, simplemente los cortamos un poco, esto ha sucedido en el montaje de varias piezas de éste proyecto).



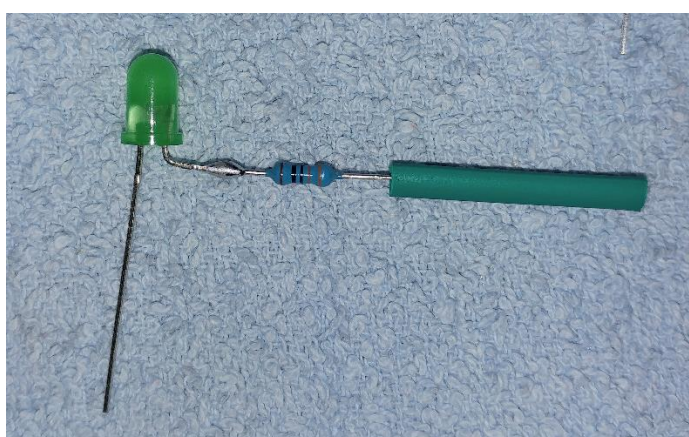
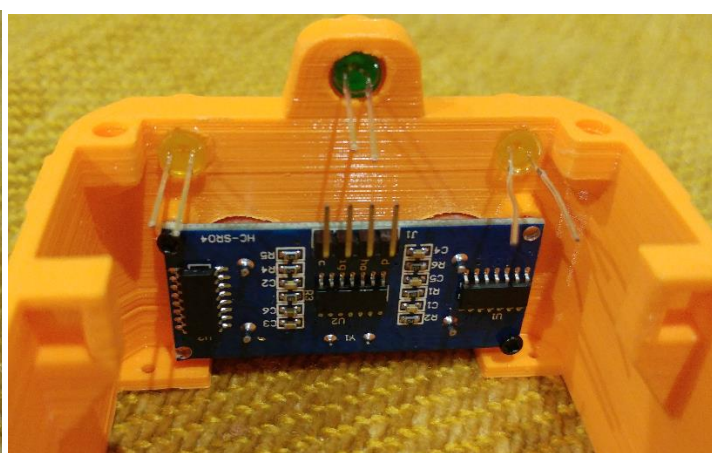
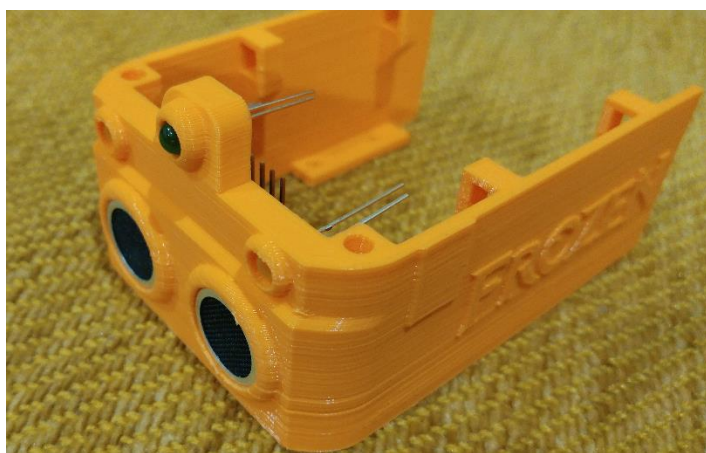
- Para la fijación del motor porta Imanes, sólo necesitamos 2 tornillos de 2mm cómo los comentados en la descripción. Tendremos que montar primero el disco con los imanes de neodimio redondos (10mm de largo x 3mm de grosor) que irán colocado a presión en el disco (quizás sería recomendable darle algo de pegamento tipo Loctite si quedan algo flojos), los imanes van colocados mirando hacia el interior de la locomotora, yo los he puesto con la misma polaridad hacia abajo.



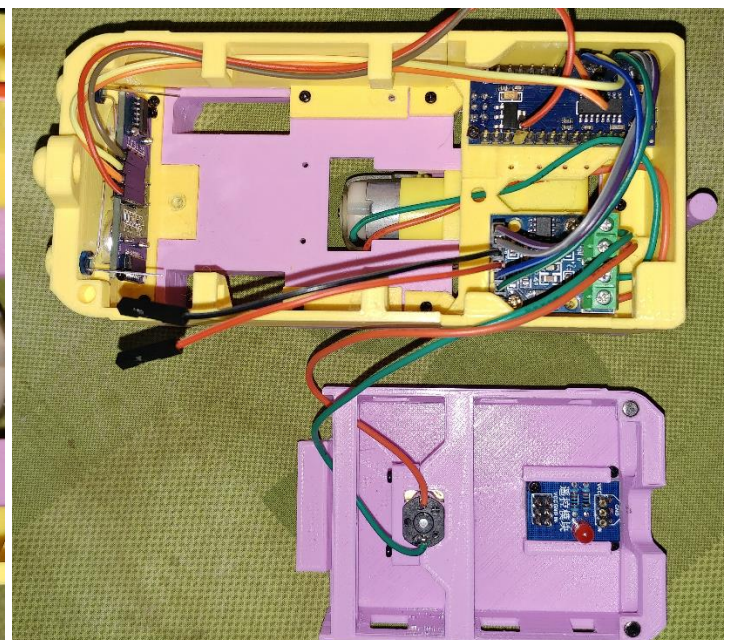
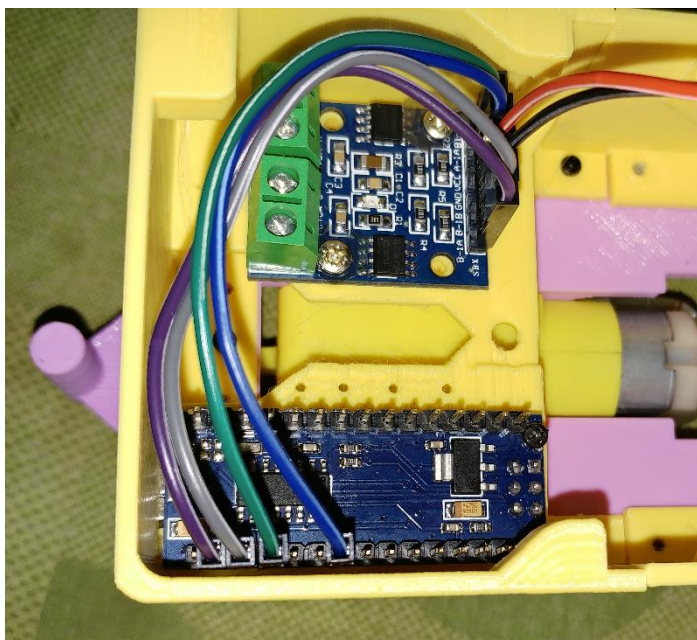
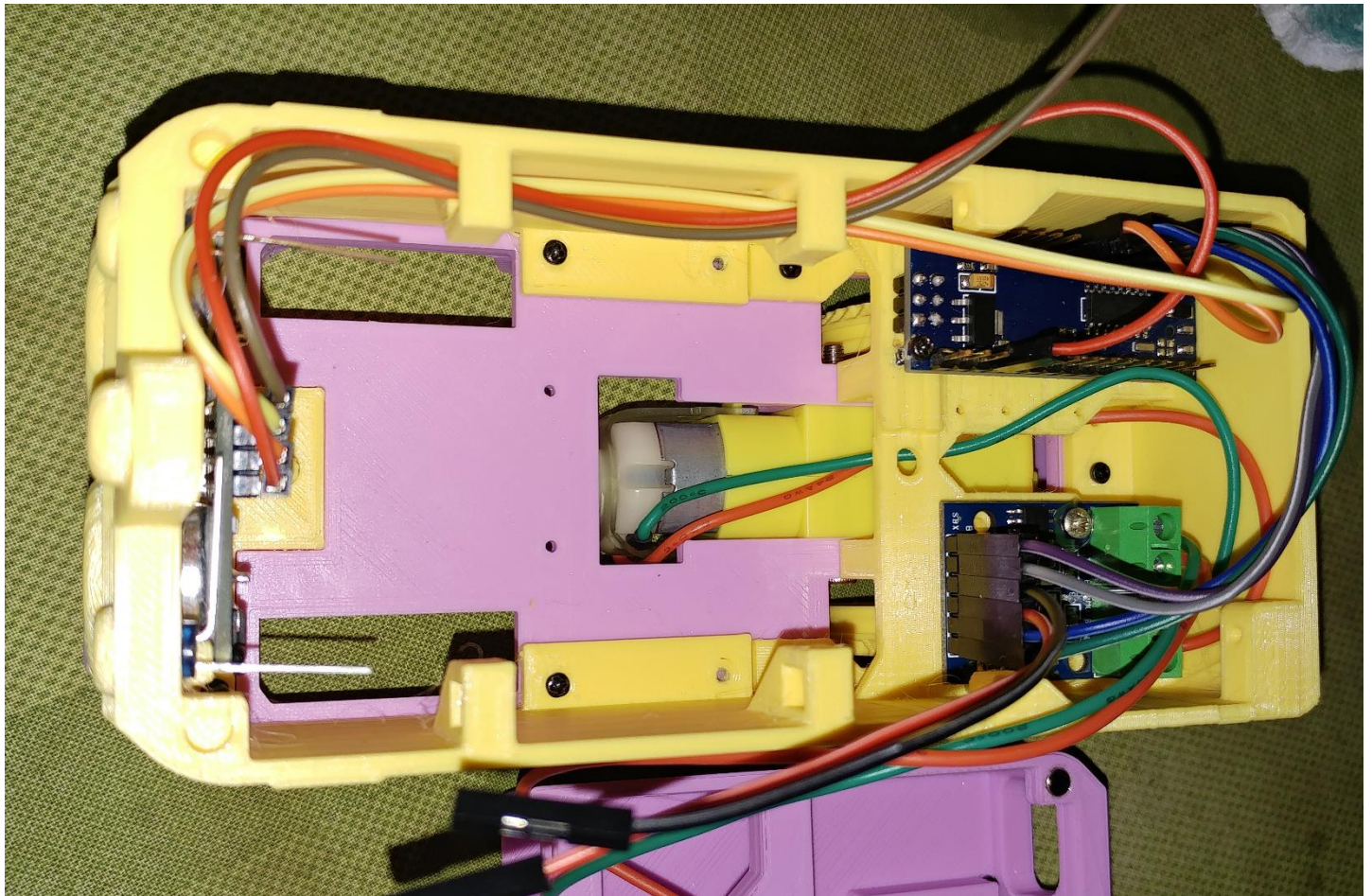
- El Arduino Nano v3.0 y el driver de control de los 2 motores van colocados en la pieza media trasera. El Arduino va encajado con las 6 patillas traseras sobre la pieza (en los huecos dejados para ello) y fijado con un tornillo de 2mm (éste quizás no encaje a la perfección). El driver puede ir sujeto con 4 tornillos en los huecos dejados para ello, (yo sólo he puesto 2 en diagonal). Esta pieza trasera va fijada al chasis hasta con 5 tornillos de 2mm.



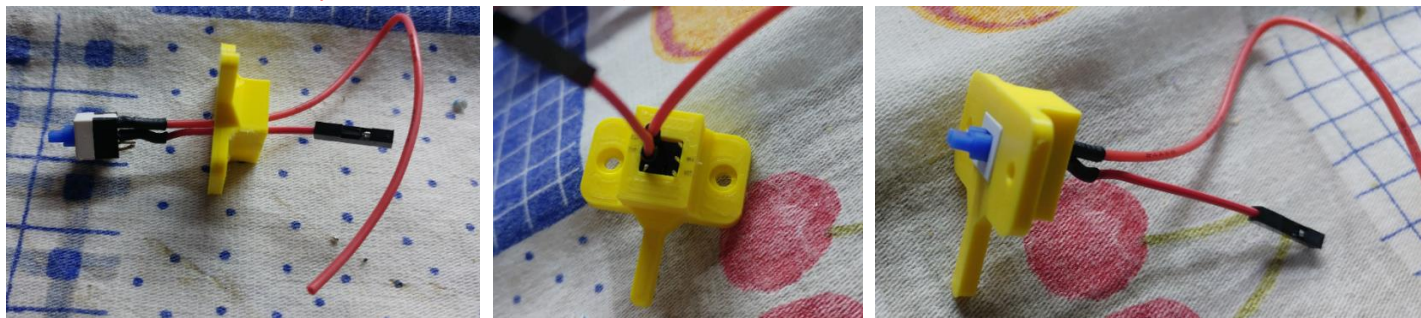
- Para la fijación del sensor de distancia por ultrasonidos usaremos 4 tornillos de 2mm, quizás tengamos que taladrar un poco más el hueco en la placa del sensor ultrasónico **HC-SR04** para que entren correctamente (yo sólo he usado 2 en diagonal, para los diodos led de las luces tendremos que usar algo de pegamento o cola térmica. Nota, podemos montar las resistencias al lado de las patillas para no ocupar mucho espacio y ponerle algo de funda termo-retráctil para esconderlas como se ve en alguna de las fotos. Podemos soldarlas a la distancia adecuada usando los huecos por la parte delantera. Ésta pieza frontal va fijada al chasis por 2 tornillos de 2mm en la parte delantera y hasta 4 en la parte trasera. Lleva 2 imanes de neodimio 4mmx4mm cilíndricos en los huecos delanteros para el cierre magnético con la parte superior.



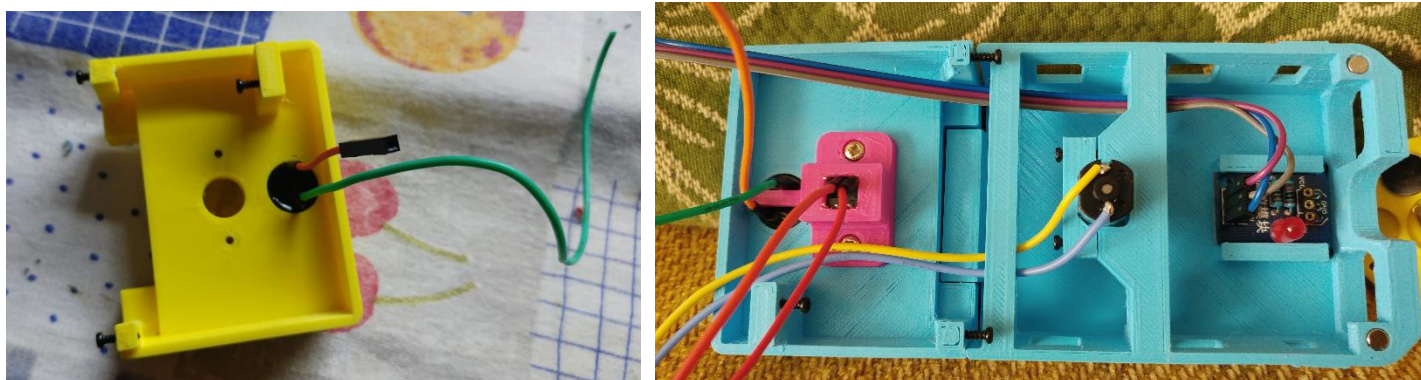
- **NOTA:** Especial atención a la hora de colocar estos imanes de cierre magnético de tapa superior con tapa inferior, **(MONTAR DE FORMA QUE SE ATRAIGAN LAS 2 PIEZAS)**, sino no conseguirás cerrar nunca las 2 piezas.
- Usaremos los huecos laterales para llevar los cables dupont hembra-hembra desde el sensor ultrasónico hasta nuestro Arduino Nano, también para todos los leds. Conectaremos los cables dupont cortos desde el driver al arduino nano, también podemos dejar conectados los cables de los 2 motores al bornero verde del driver. **Tener en cuenta que la alimentación (VCC) del sensor de distancia por ultrasonidos va a la patilla de 5V de nuestro Arduino Nano.**



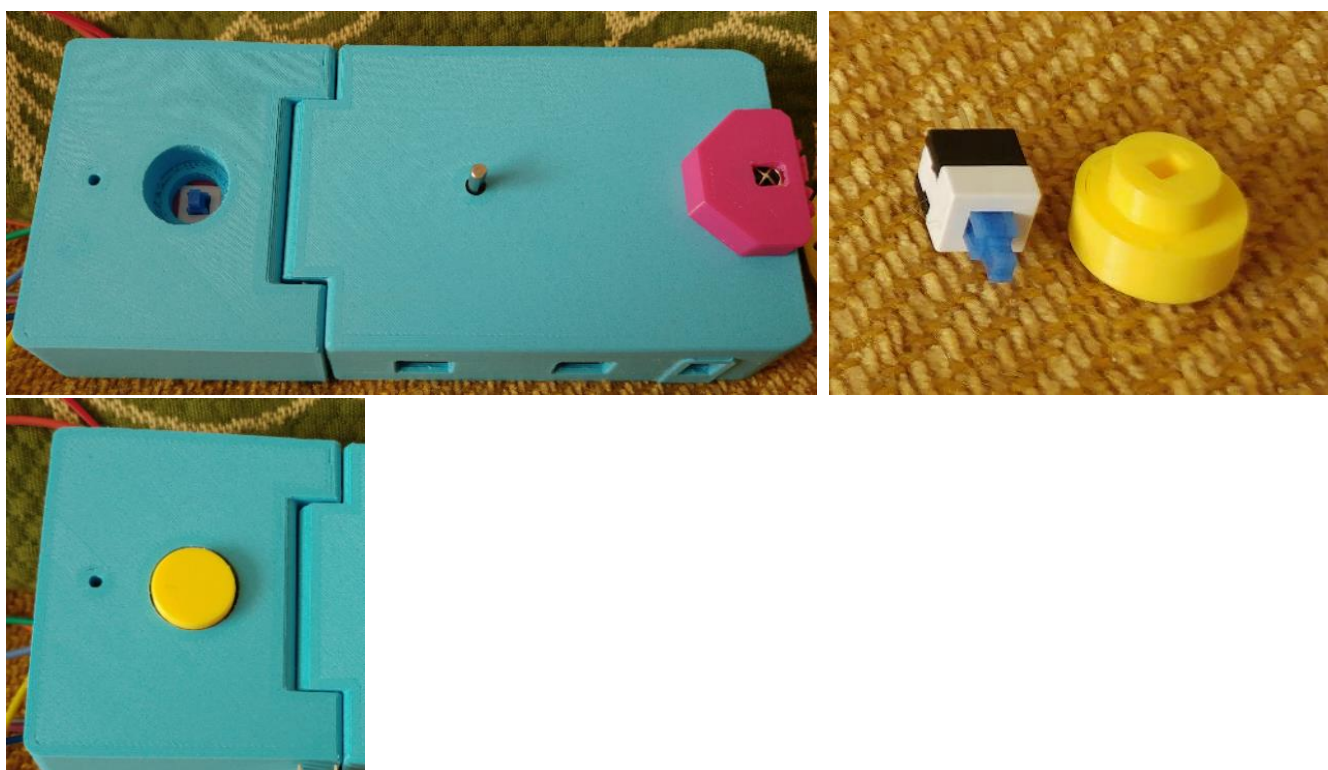
- Colocaremos el interruptor (tipo pulsador) de encendido en su pieza que irá fijada con tornillería de 2mm a la pieza superior trasera (ésta pieza también evita que el altavoz se salga de su alojamiento). En éste dejaremos uno de los hilos con cable dupont Hembra para llevar junto con la alimentación del driver a la entrada de Vin del Arduino, el otro hilo irá directamente soldado al terminal positivo de la batería. **Nota: Mejor dejar 2 hilos con cable dupont hembra para llevar a la Vin de Arduino y VCC del driver HG7881 (L9110S), (EN LAS FOTOS SÓLO APARECE UNO).**



- Dejaremos el altavoz ya soldado, éste tiene polaridad, fijarse bien cual es el hilo positivo ya que lo dejaremos con conector dupont hembra y el negativo, que quedará unido a los pines GND del Arduino, junto con negativo de la batería, negativo del sensor infrarrojo, negativo del sensor ultrasonido, negativo del driver HG7881 (L9110S), negativo del Arduino y de los 3 diodos leds. Esta pieza lleva 3 tornillos de 2mm para su fijación con la pieza media trasera (2 tornillos son necesario recortarlos un poco para que no sobresalgan). **Las dos piezas superiores van unidas por una varilla de 2mm de grosor y aproximadamente de 51mm de largo.** Luego lleva una pequeña pieza redonda que hace de tapón.



- Para el montaje de la pieza que va sobre el interruptor (pulsador de encendido) simplemente daremos una gotita de pegamento tipo loctite en la parte de arriba del pulsador **una vez montado todo en sus sitio**, para fijarla. No hechar demasiado ya que podríamos estropear el pulsador.



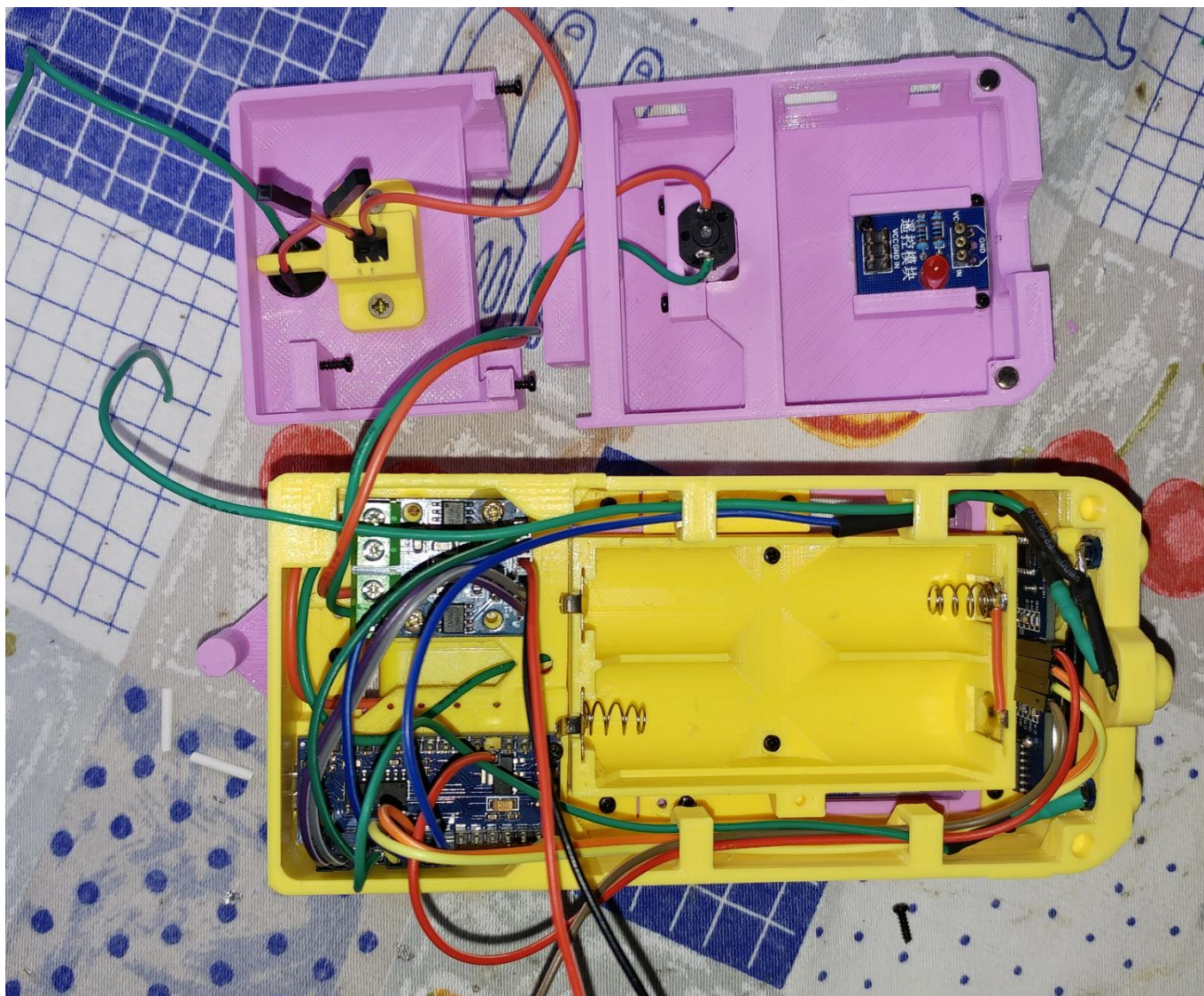
- En la batería tendremos que unir los terminales de uno de los lados (el terminal macho y el que lleva muelle, por la parte que se encuentra algo rebajada “hilo verde en la fotografía”) y dejaremos los terminales de conexión positivo y negativo hacia la parte de atrás de la locomotora, deberemos plegarlos un poco para que no moleste al poner la tapa de la batería. Ésta está pendiente de recortar un poco para que no tropiece con los terminales del sensor de ultrasonidos ya que entra muy justa. Va fijada al chasis con 2 tornillos de 2mm.



- Si quisieramos ir a lo más cómodo también podemos optar a comprar la caja de baterías de litio 18650, algo como esto (tendríamos que taladrar el chasis o la caja para su fijación), algo como esto:
- <https://www.ebay.es/itm/Caja-pl%C3%A1stico-de-almacenamiento-Soporte-carga-bater%C3%ADa-18650-recargable-cable/333115997405?ssPageName=STRK%3AMEBIDX%3AIT&var=542141549431&trksid=p2060353.m2749.l2649>



- Bueno, esto es un poco por encima la explicación del montaje, tener en cuenta que el arduino está montado con los terminales de conexión hacia arriba y no se observa que es cada pin ya que la serigrafía está por la otra cara de la placa, por lo que es sencillo equivocarse. Revisar con lupa el montaje una vez finalizado para que todo trabaje correctamente.



- **CARGA DE NUESTRO PROGRAMA EN ARDUINO NANO:**

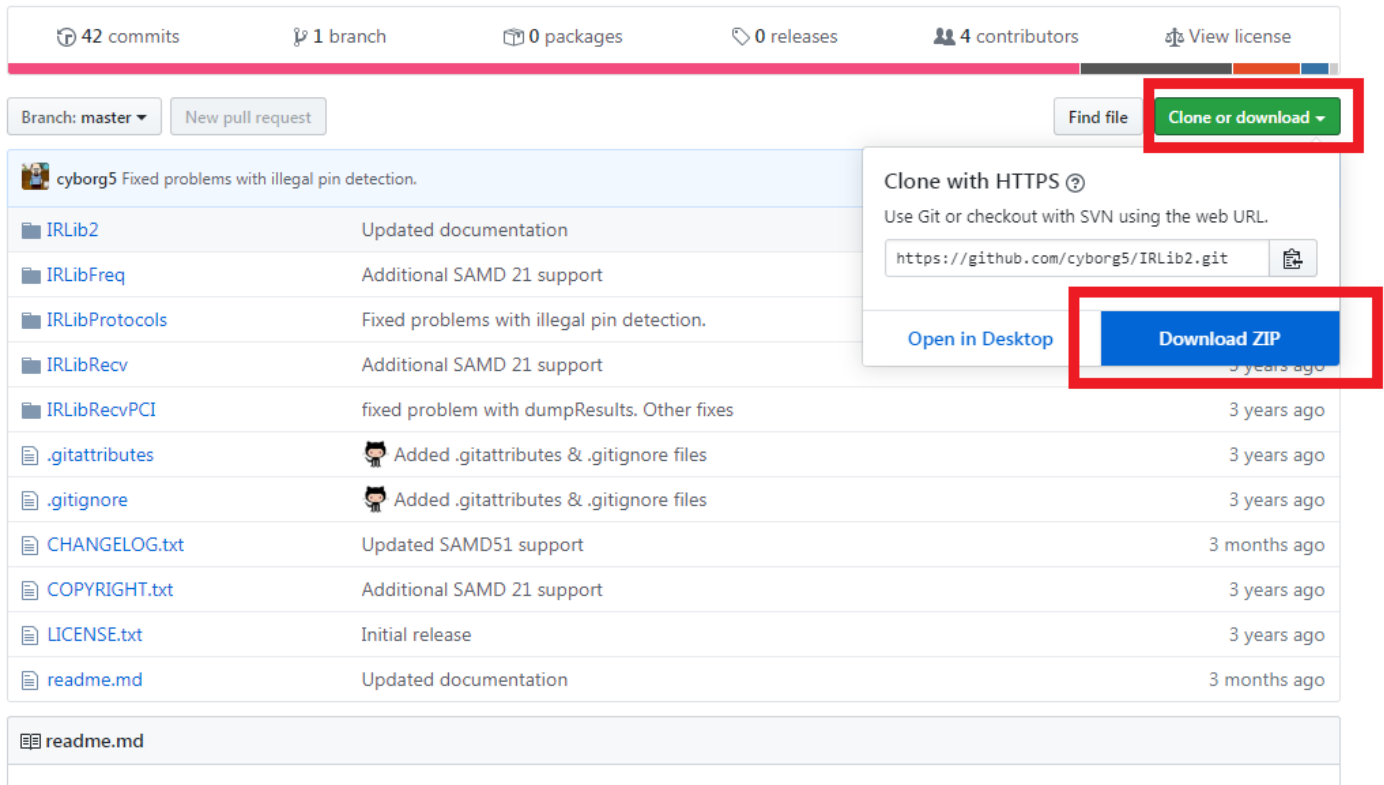
- **Nota:** La carga de nuestro programa, sería recomendable hacerlo primeramente antes de tener todo montado y conectado, así ya tendríamos las entradas y salidas ya definidas con las que vamos a trabajar (aunque no habría ningún problema, el no hacerlo).
- Primeramente descargamos e instalamos el software Arduino Nano en caso de no tenerlo instalado de ésta dirección: <https://www.arduino.cc/en/main/software>

Download the Arduino IDE



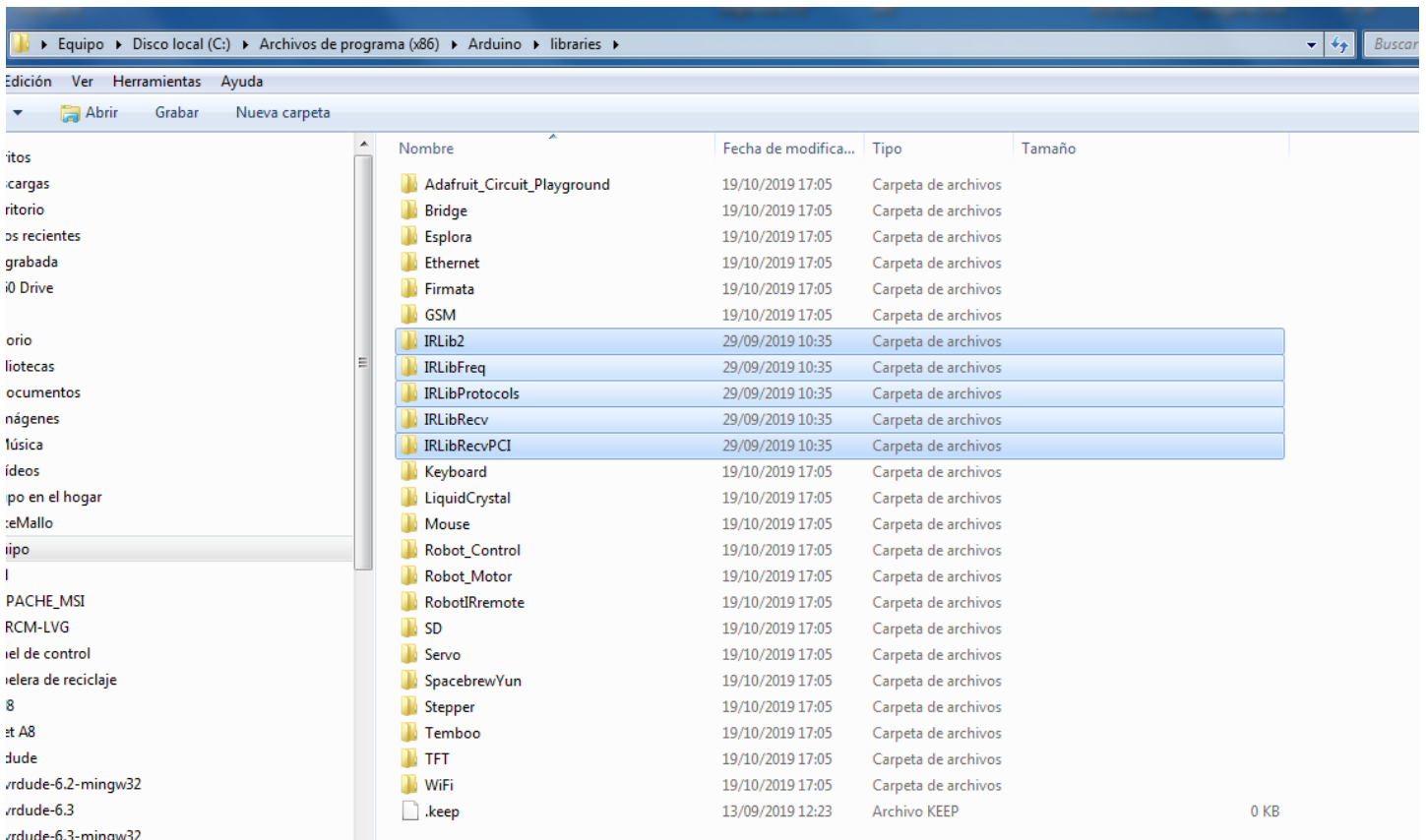
- También debemos de instalar la librería “**IRLIB2**” que se incluye en el proyecto para que pueda trabajar correctamente nuestro mando infrarrojo, podemos descargarla de ésta dirección: <https://github.com/cyborg5/IRLib2>

Library for receiving, decoding, and sending infrared signals using Arduino



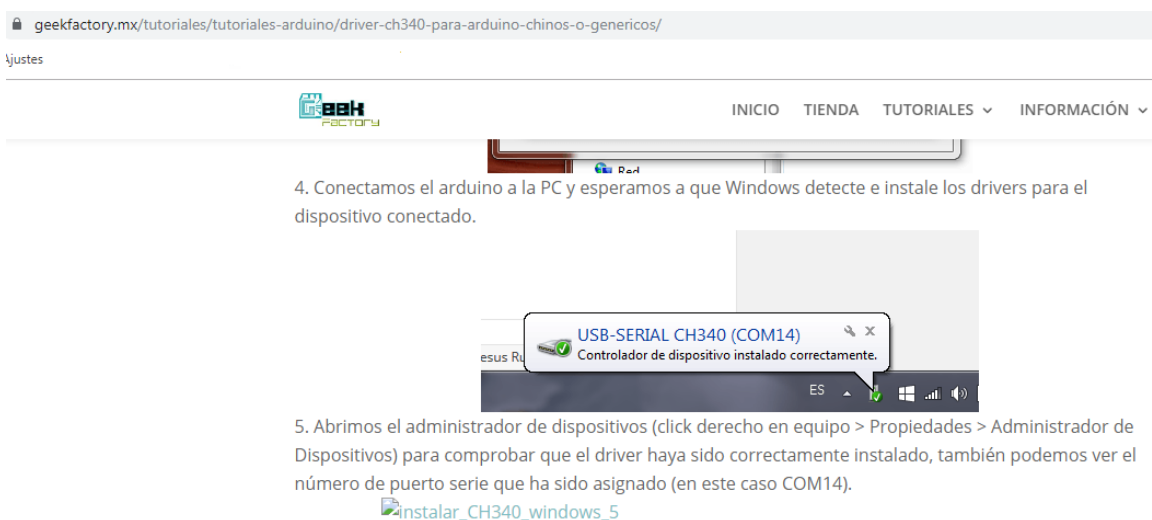
Commit	Message	Time
cyborg5	Fixed problems with illegal pin detection.	
IRLib2	Updated documentation	
IRLibFreq	Additional SAMD 21 support	
IRLibProtocols	Fixed problems with illegal pin detection.	
IRLibRecv	Additional SAMD 21 support	
IRLibRecvPCI	fixed problem with dumpResults. Other fixes	3 years ago
.gitattributes	Added .gitattributes & .gitignore files	3 years ago
.gitignore	Added .gitattributes & .gitignore files	3 years ago
CHANGELOG.txt	Updated SAMD51 support	3 months ago
COPYRIGHT.txt	Additional SAMD 21 support	3 years ago
LICENSE.txt	Initial release	3 years ago
readme.md	Updated documentation	3 months ago

- De todos modos la incluiré en la pagina de Thingiverse para que podáis descargarla fácilmente, nombre de la carpeta **"IRLib2-master"**.
- **Tenemos que descomprimir las carpetas de ésta librería y copiarlas dentro de la carpeta (Arduino/libraries) quedando cómo se ve en la imagen.**
- La carpeta principal "IRLib2-master" dónde estaban contenidas el resto de carpetas "IRLib2, IRLibFreq, ..." se elimina.



- Nuestro Arduino nano (versión china) lleva el chip CH340 para la comunicación mediante cable USB, por lo tanto también tenemos que tener instalado los drivers para poder comunicar correctamente. Podemos descargarlos de ésta dirección (**los enlaces se encuentran al final de la página web**):

<https://www.geekfactory.mx/tutoriales/tutoriales-arduino/driver-ch340-para-arduino-chinos-o-genericos/>

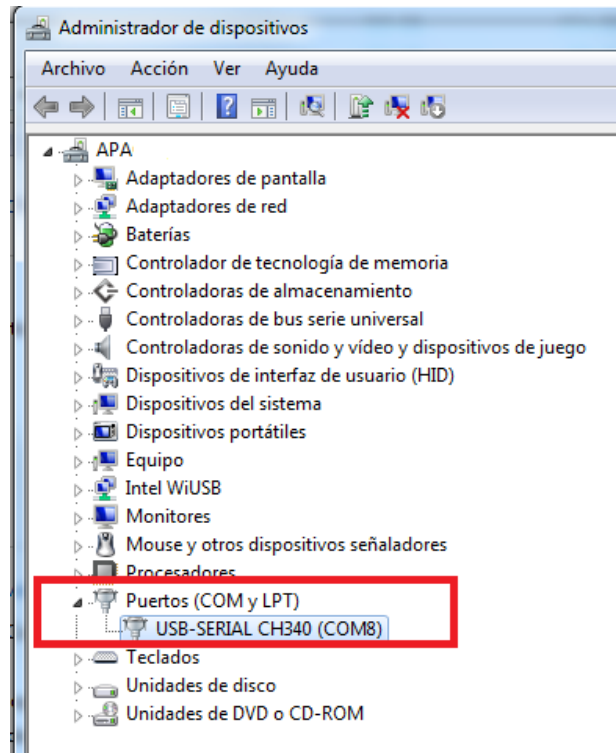


Descarga del driver CH340 para arduino chinos o genéricos

A continuación los enlaces de descarga para el **driver CH340**, versiones más actualizadas (Noviembre 2017):

- [Descarga del driver para Windows](#)
- [Descarga del driver para MAC](#)
- [Descarga del driver para Linux](#)

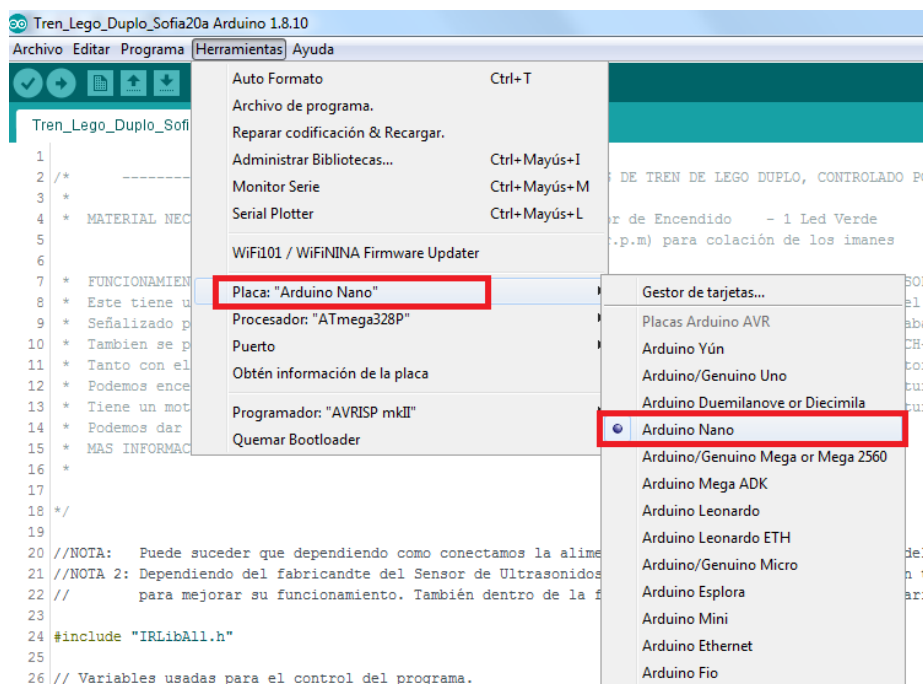
- Una vez **instalado los drivers y conectado mediante un cable USB** de nuestro ordenador al Arduino Nano (no es necesario tener nada más conectado a nuestro Arduino Nano) podemos observar el **puerto** que utiliza para la comunicación entrando en el **“Administrador de Dispositivos”** dentro de nuestro Windows. Posteriormente debemos seleccionarlo en el software Arduino para poder mandarle nuestro programa.



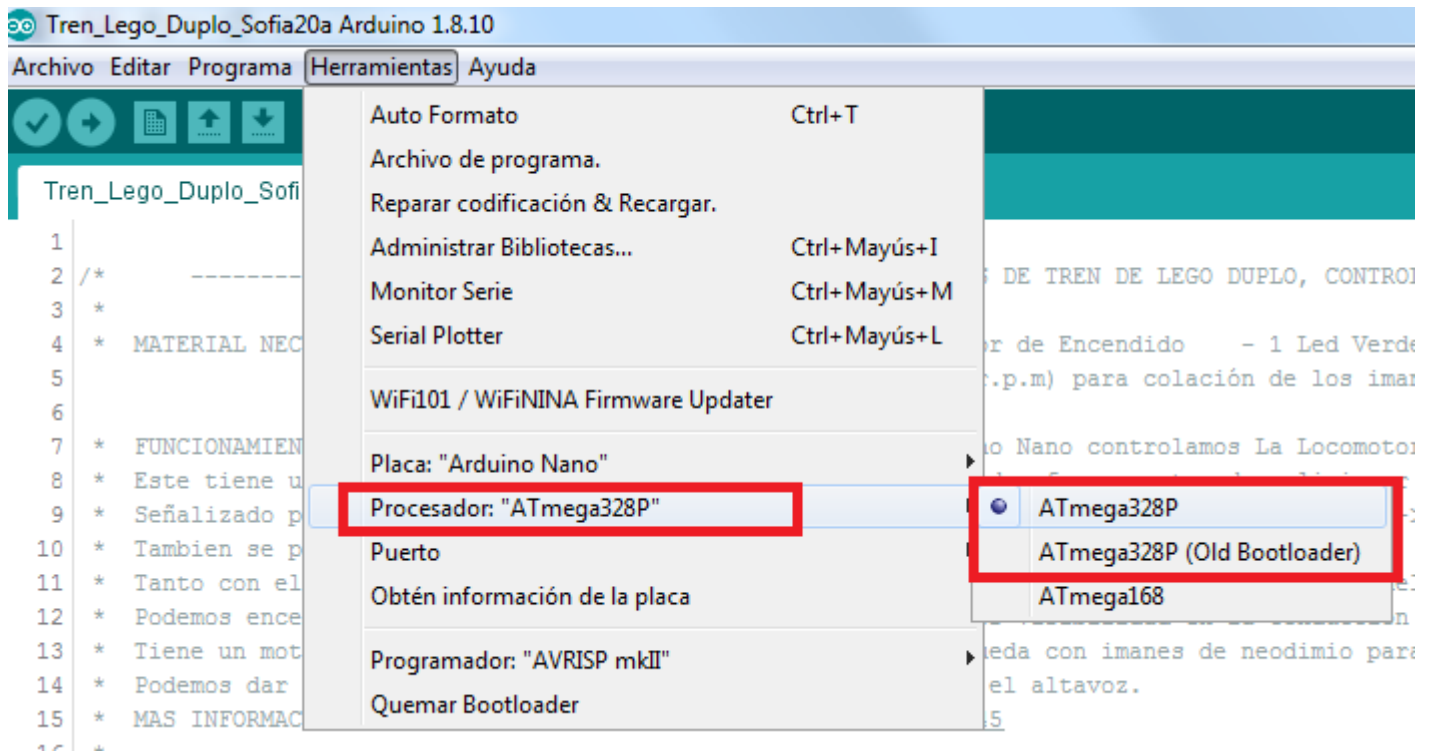
- Bien finalmente descargamos el programa que vamos a mandarle a nuestro Arduino Nano **“Tren_Lego_Duplo_Sofia_v1.2.ino”** (debe de estar contenido en una carpeta con el mismo nombre)



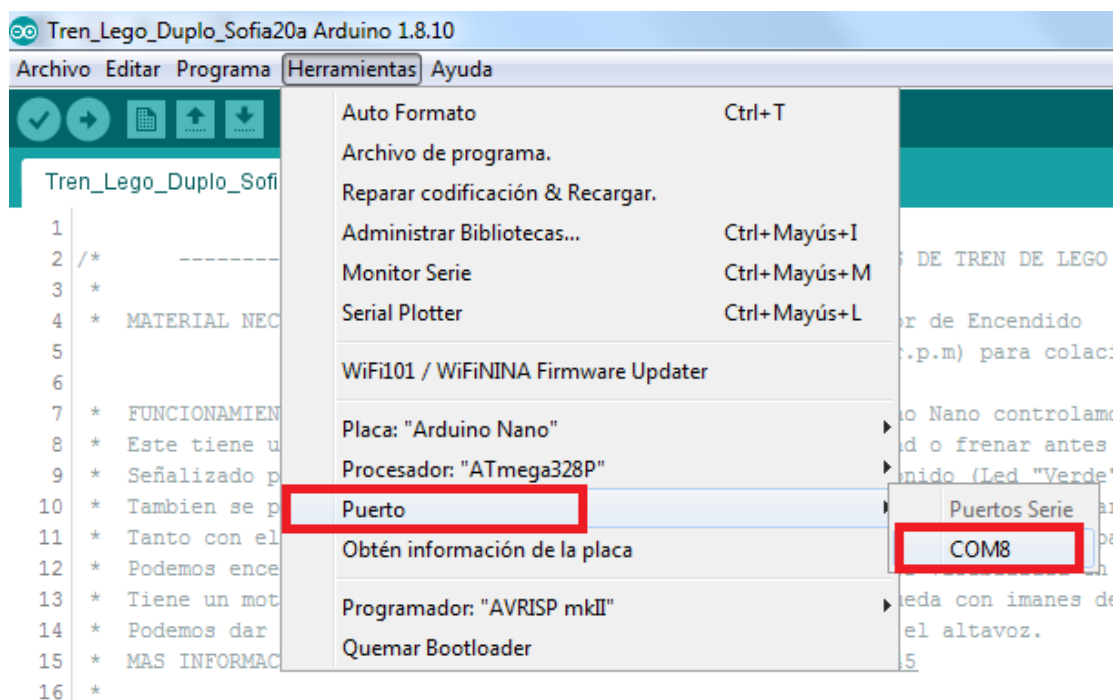
- Una vez tenemos todo lo necesario ya descargado y conectado, arrancamos el Software Arduino, vamos a la pestaña **“Herramientas”** y **“Placa”** y seleccionamos **“Arduino Nano”**





- Ahora dentro de la misma pestaña de “Herramientas”, seleccionamos “Procesador” y luego seleccionamos “ATmega328P” .
- Bien aquí también podemos seleccionar “ATmega328P(Old Bootloader)”, en caso de que no consigamos comunicar bien con la primera opción. A mi me ha pasado con dos diferentes Arduino Nano, tener que seleccionar uno u otro para que la comunicación fuera correcta. Se selecciona el primero, si tarda demasiado en subir el programa, debemos probar con el otro.



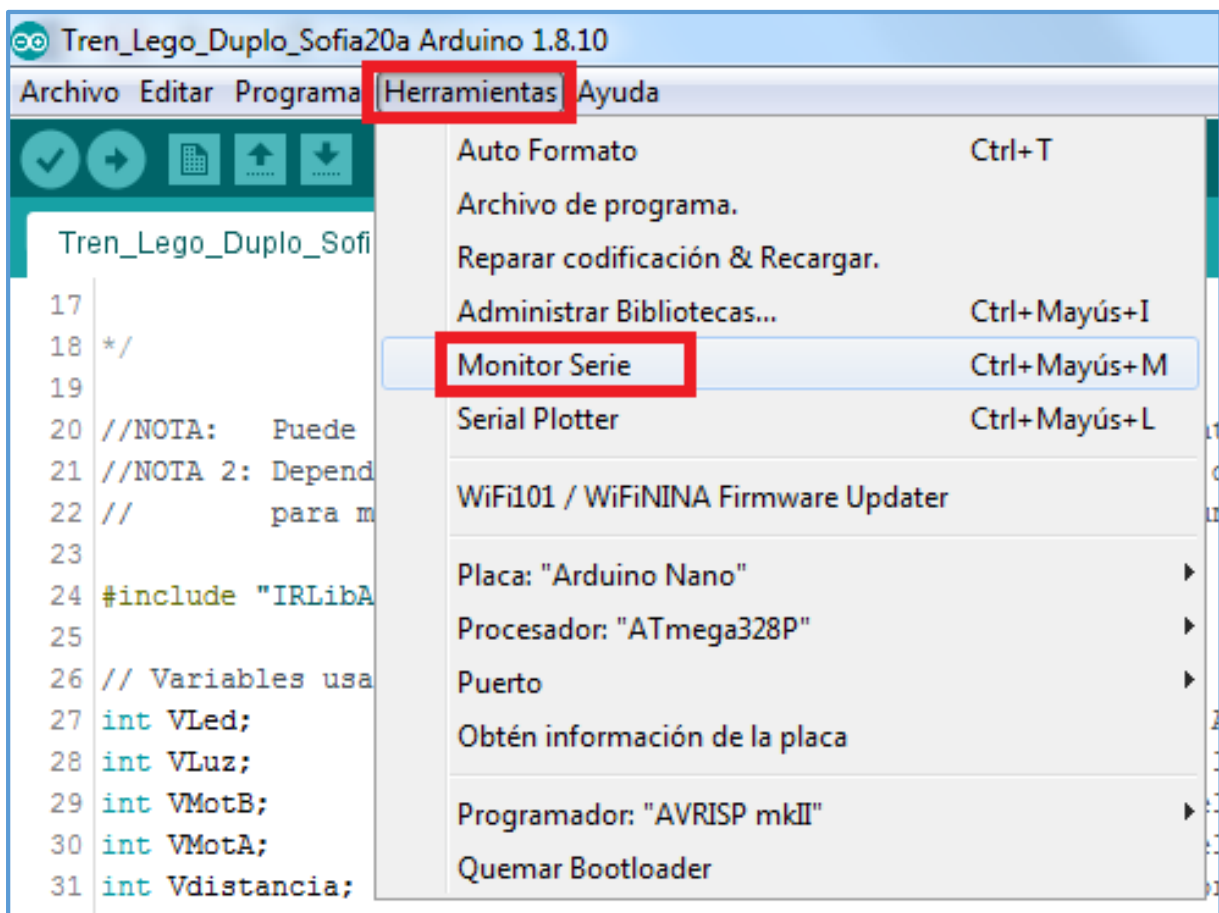
- Ahora dentro de ésta pestaña de “Herramientas” escogemos “Puerto”, y seleccionamos el puerto que nos aparecía en el “Administrador de Dispositivos” en mi caso “COM8” y con esto tenemos listo la configuración para poder Subir nuestro programa.



- Abrimos el archivo “Tren_Lego_Duplo_Sofia_v1.2.ino”, y le damos a la tecla de “Verificar”  para comprobar que no hay ningún error en el programa y posteriormente le damos a “Subir” . Si todo está correcto debería de salir el mensaje de subida y los datos de espacio libre, bytes usados....

```
Subido
El Sketch usa 9122 bytes (29%) del espacio de almacenamiento de programa. El máximo es 30720 bytes.
Las variables Globales usan 476 bytes (23%) de la memoria dinámica, dejando 1572 bytes para las variables locales. El máximo es 2048 bytes.
```

- Una vez todo conectado, y con nuestro programa cargado, al darle alimentación a la locomotora (con pulsador de encendido) debería de encenderse el led “verde” (central) de trabajo con sensor de distancia por ultrasonidos activado (varía la velocidad del tren hasta pararse si está cerca de colisionar).
- Si tenemos el mando por infrarrojo recomendado en la descripción del material (**kit VS1838**) podemos probar todos los botones para su correcto funcionamiento.
- **Nota:** En caso de que el motor de las ruedas giren en sentido contrario a lo que sale en la descripción del mando a distancia (“CH+” = **Avance de la Locomotora**), lo más sencillo es intercambiar los 2 cables que van a ese motor en el bornero “verde” (con tornillos) del driver **HG7881 (L9110S)**. Esto es importante ya que el Control de Velocidad mediante el sensor de distancia por ultrasonidos sólo trabaja en el sentido de avance del tren (lo normal ya que es donde tenemos montado nuestro sensor ultrasónico), cuando estamos marcha atrás pues evidentemente no tenemos otro sensor que pueda evitar colisiones. Si sucede que están invertidos éstos botones de avance y retroceso del mando (“CH+” y “CH-”) lo que pasaría es que el sensor de ultrasonidos trabajaría cuando el tren va marcha atrás (evidentemente, mal funcionamiento).
- **En caso de tener otro mando infrarrojo** (se puede utilizar el de un televisor, si lo soporta la librería) debemos configurar cada tecla de nuestro nuevo mando y modificar el programa del Arduino para que trabaje correctamente.
- Para hacer ésto una vez subido nuestro programa, nos vamos a “Herramientas” y Seleccionamos “Monitor Serie”.



- Nos saldría **una nueva ventana** en la cual, después de pulsar una tecla de nuestro mando infrarrojo, nos genera una valor en hexadecimal (después de **Value:**), en éste ejemplo **"FF38C7"** (Sin la comillas), que es el que vamos a tener que usar para modificar en nuestro programa.

- **Nota:** Si tenemos el **"control de Velocidad por Ultrasonidos Activo"** (señalizado por led verde encendido), **tendremos que apagarlo** para que no esté refrescando continuamente el valor de velocidad actual y distancia que esta midiendo el sensor ultrasónico en la pantalla de Monitor Serie. Pulsamos Tecla **"Prev"** a **Anterior** del mando a distancia, (se tiene que apagar el led verde). También podemos hacerlo despinchando la casilla **"Autoscroll"** en esta ventana.

```
Decoded NEC(1): Value:FF38C7 Adrs:0 (32 bits)
Raw samples(68): Gap:5562
  Head: m8890 s4550
0:m510 s614    1:m514 s614    2:m510 s614    3:m514 s614
4:m510 s614    5:m510 s618    6:m510 s614    7:m510 s618
8:m510 s1714   9:m510 s1742   10:m486 s1718  11:m510 s1718
12:m510 s1718  13:m514 s1714  14:m510 s1718  15:m510 s1742

16:m486 s638   17:m490 s618   18:m506 s1742  19:m486 s1718
20:m510 s1718  21:m514 s614   22:m510 s638   23:m490 s614
24:m510 s1742  25:m486 s1718  26:m510 s618   27:m510 s614
28:m510 s638   29:m486 s1718  30:m510 s1718  31:m510 s1718
```

- Dentro de el programa de Arduino buscamos la rutina del mando infrarrojo, llamada **"translateIR()"**, las diferentes teclas del mando infrarrojo se configuran en la instrucción **"switch"** y el valor generado anteriormente va situado después de cada **"case"**, que tendremos que modificar para nuestro nuevo mando. Se le añade el valor **"0x"** (Sin la comillas) quedando de ésta forma **case 0xFF38C7:** (para éste ejemplo). Dependiendo en que **"case"** lo pongamos pues nos hará una cosa u otra, va comentado en el programa de Arduino que sucede en cada opción.

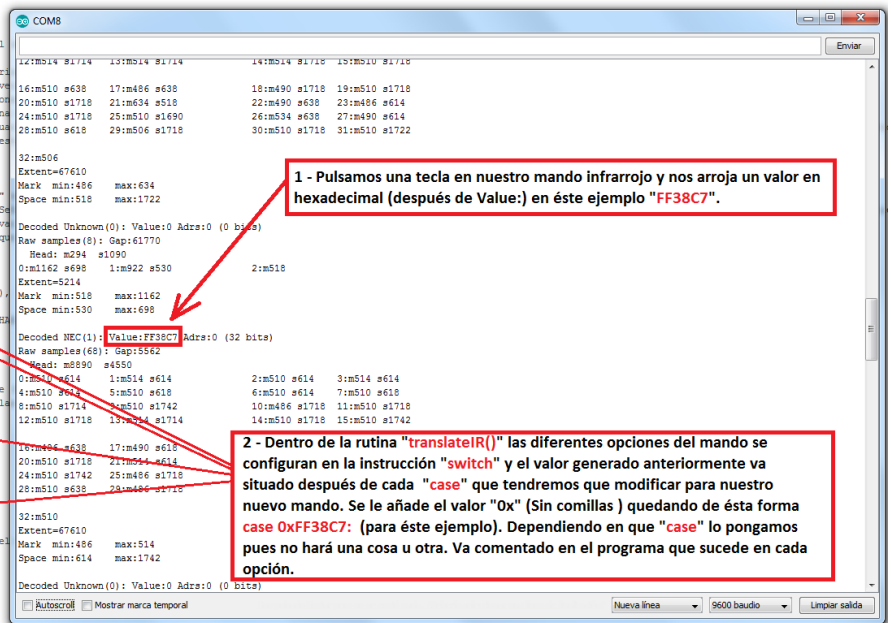
```
22 void translateIR()
23 {
24     switch(myDecoder.value) {
25         case 0xFF22DD: // El valor "FF22DD" es la tecla
26             if(VLed==0) { // SE ACTIVA O DESACTIVA EL "LED"
27                 digitalWrite(Led1,HIGH); // Activamos el Pin "D3" usado
28                 VLed=1; // Valor estado de la variable
29                 buzzer();
30             }
31             else{
32                 digitalWrite(Led1,LOW); // En caso contrario (que VLed
33                 VLed=0;
34                 VMotB=0; // PENDIENTE DE PROBAR SI NO ARRANCA DESPUES DE ESTAR PA
35                 buzzer2();
36             }
37         break;
38
39         case 0xFF02FD: // El valor "FF02FD" es la tecla
40             if(VLuz==0) { // La rutina básicamente trab
41                 digitalWrite(luz,HIGH);
42                 VLuz=1;
43             }
44             else{
45                 digitalWrite(luz,LOW);
46                 VLuz=0;
47             }
48         break;
```


Tren_Lego_Duplo_Sofia20a

```

211
212 //MANDO DE INFRARROJOS
213 // Dependiendo del código recibido por infrarrojo se ejecuta las diversas acciones:
214 // - Activación/Desactivación de la luz de encendido, (led verde) que señaliza que el control
215 // - Activación/Desactivación de las Luces delanteras (2 leds blancos)
216 // - Motor A (Rortaimanes) Giro a derechas o izquierdas (Varias Velocidades, si pulsamos varia
217 // - Motor B (Ruedas) Adelante/Atrás (Si pulsamos varias veces en el mismo sentido sube de ve
218 // - Sonidos Activados/Desactivados (trabaja el altavoz activo para señalizar diversas acción
219 // - NOTA: En caso de querer utilizar un mando infrarrojo diferente (quizás valdría uno de una
220 // la palabra "Value:FFFFFF", Estas "FFFFFF" es el código que tendrías que poner a continua
221 // Nota: al estar activo el Sensor de Ultrasonido para poder observar el código obtenido des
222 void translateIR()
223 {
224   switch (myDecoder.Value) {
225     case 0xFF22DD: // El valor "FF22DD" es la tecla "Anterior"
226     if (VLed==0) // SE ACTIVA O DESACTIVA EL "LED VERDE" "Se
227       digitalWrite(Led1,HIGH); // Activamos el Pin "D3" usado para activar
228       VLed=1; // Valor estado de la variable activo ya q
229       buzzer();
230     }
231     else{
232       digitalWrite(Led1,LOW); // En caso contrario (que VLed esté a 1),
233       VLed=0;
234       VMotB=0; // PENDIENTE DE PROBAR SI NO ARRANCA DESPUES DE ESTAR PENSADO AL DESHA
235       buzzer2();
236     }
237     break;
238
239     case 0xFF02FD: // El valor "FF02FD" es la tecla "Siguiente"
240     if (VLuz==0) // La rutina básicamente trabaja que si la
241       digitalWrite(Luz,HIGH);
242       VLuz=1;
243     }
244     else{
245       digitalWrite(Luz,LOW);
246       VLuz=0;
247     }
248     break;
249
250     case 0xFFE01F: // El valor "FFE01F" es la tecla "-" en el
251     if (VMotA !=1){
252       Speed_MA = FPM_SLOW_A;
253       go_forward_A ();
254     }
255     else{
256       Speed_MA = Speed_MA + SpeedStep_MA ;

```



```

COM8
14:mS14 B1714 15:mS14 B1714 14:mS14 B1715 15:mS14 B1715
16:mS10 e638 17:m486 e638 18:m490 e1718 19:mS10 e1718
20:mS10 e1718 21:m634 e518 22:m490 e638 23:m486 e614
24:mS10 e1718 25:mS10 e1690 26:m534 e638 27:m490 e614
28:mS10 e618 29:m506 e1718 30:mS10 e1718 31:mS10 e1722

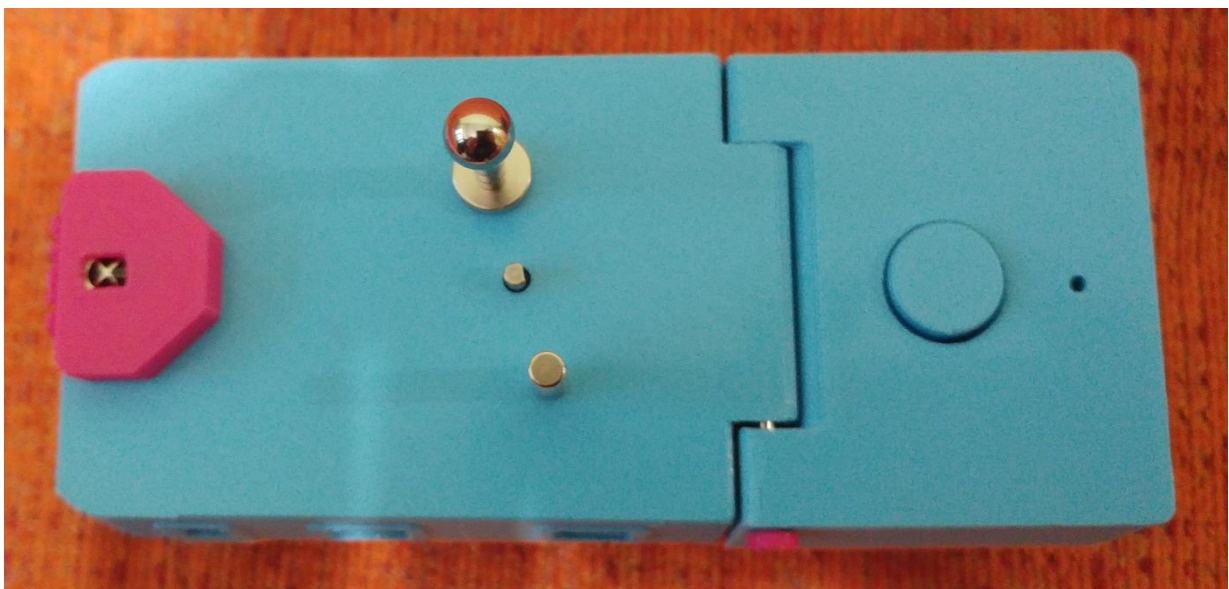
32:m506
Extent:67610
Mark min:486 max:634
Space min:518 max:1722
Decoded Unknown(0): Value:0 Adre:0 (0 bits)
Raw samples(8): Gap:61770
Head: m294 s1090
0:m162 e698 1:m822 e530 2:m518
Extent:5214
Mark min:518 max:1162
Space min:530 max:698
Decoded NEC(1): Value:FF38C7 Adre:0 (32 bits)
Head: m5890 s4550
0:mS10 e614 1:mS14 e614 2:mS10 e614 3:mS14 e614
4:mS10 e614 5:mS10 e618 6:mS10 e614 7:mS10 e618
8:mS10 e1714 9:mS10 e1742 10:m486 e1718 11:mS10 e1718
12:mS10 e1718 13:m244 e1714
14:mS10 e1718 15:mS10 e1742
16:m486 e638 17:m490 e638
20:mS10 e1718 21:m244 e614
24:mS10 e1742 25:m486 e1718
28:mS10 e638 29:m486 e1718

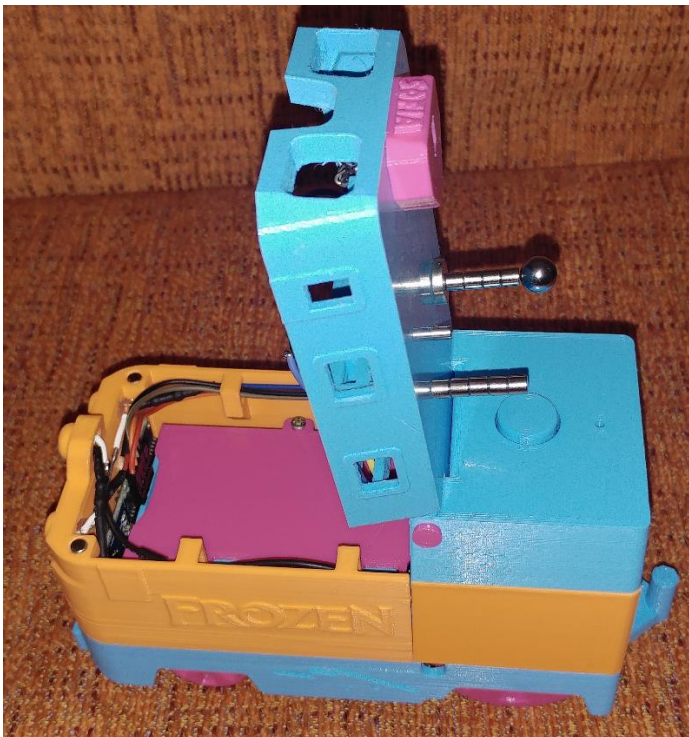
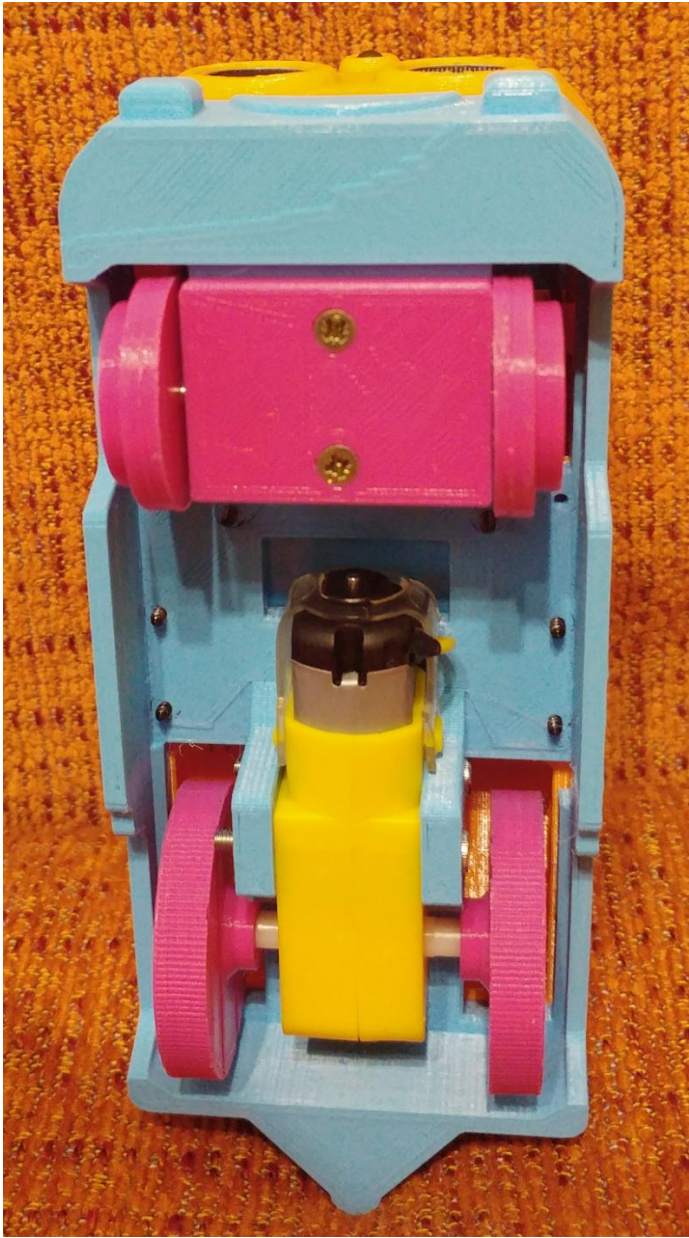
32:m510
Extent:67610
Mark min:486 max:514
Space min:614 max:1742
Decoded Unknown(0): Value:0 Adre:0 (0 bits)
Autoscroll Mostrar marca temporal
Nueva línea 9600 baudio Limpiar salida

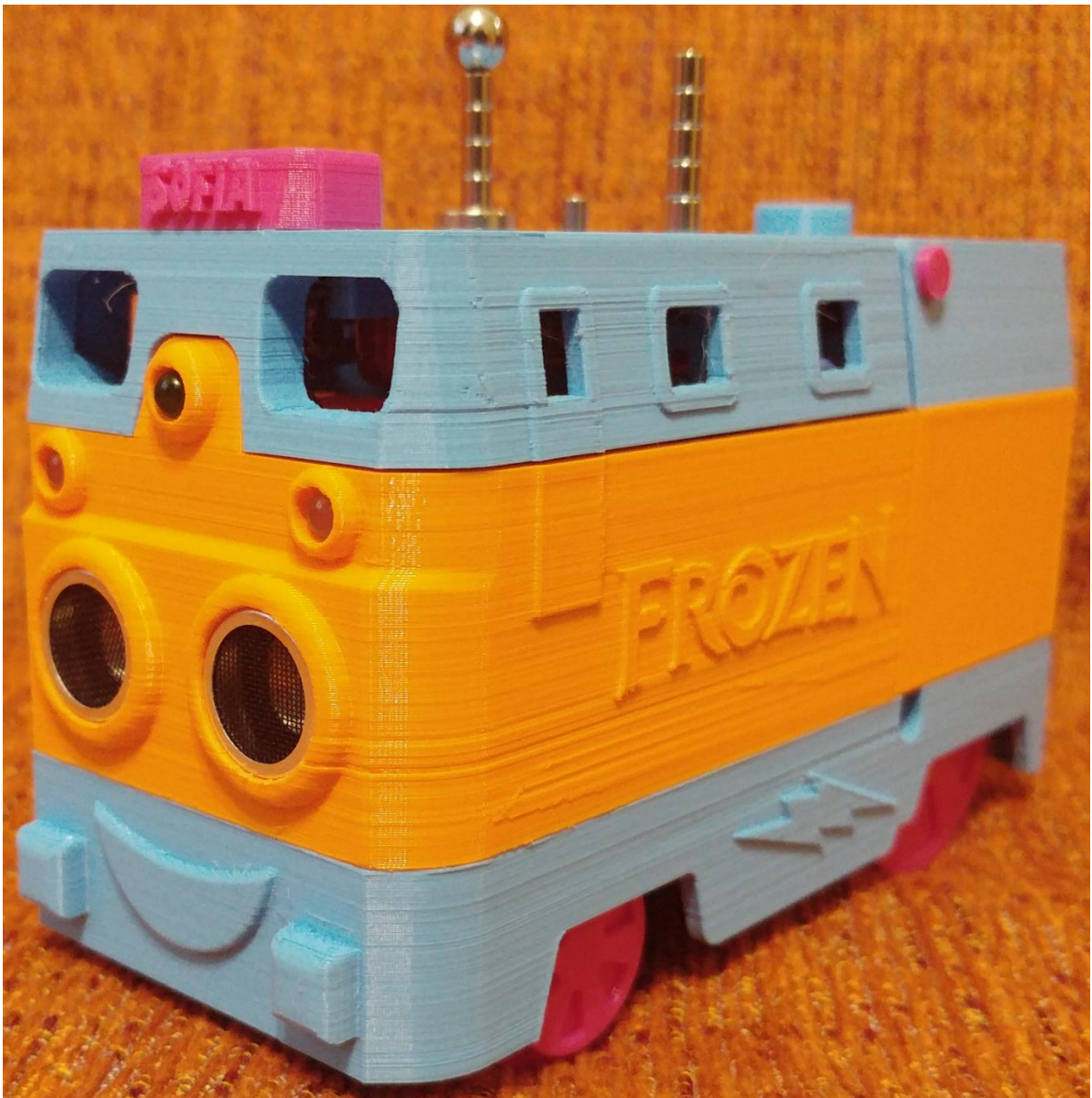
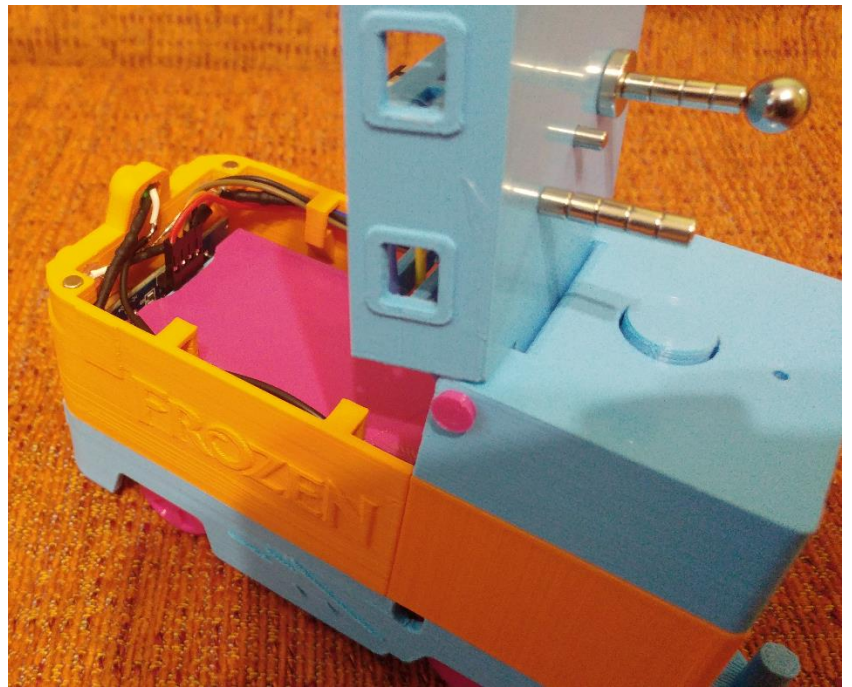
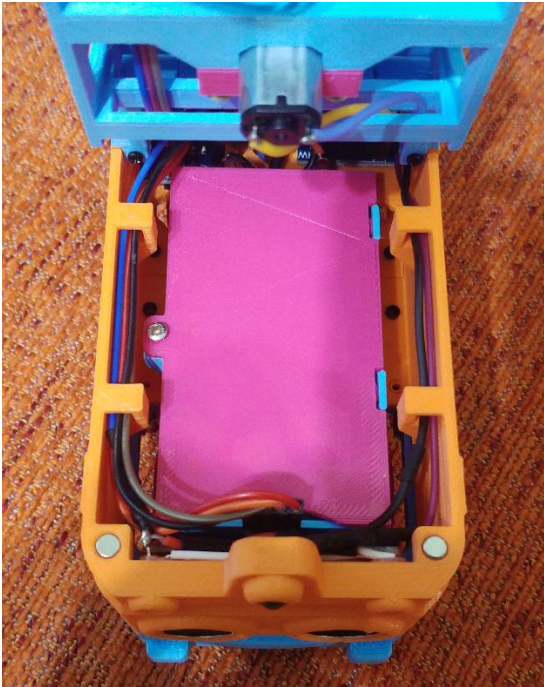
```

- Cuando tenga algo más de tiempo para hacer pruebas quizás haga modificaciones en el programa para mejorar el comportamiento del sensor de distancia por ultrasonidos. Tener en cuenta que éste modelo de sensor tiene un rango de hasta unos 5 metros de detección (aunque a partir de los 4 metros se vuelve impreciso según fabricante) **y un ángulo de detección de 15 grados**. Éste ángulo nos limita por ejemplo a la hora de detectar otro tren que va por la misma vía en una curva, ya que el sensor siempre va controlando la distancia de frente, por lo que siempre nos lo va a detectar, cuando ya esté casi pegado, **llegando a chocar ligeramente en muchas ocasiones**. Tampoco nos serviría en caso de un cruce por la misma razón.
- De todos modos al tener la opción de control manual pudiendo subir y bajar la velocidad siempre se hace algo más divertido para los peques y no tan peques (**modo ultrasonidos desactivado**).
- Normalmente una vez que cambiamos el sentido de giro de nuestra locomotora (led verde apagado), éste se para y comienza con una velocidad media. Esta velocidad media también, es el tope programado cuando tenemos el control de velocidad por ultrasonidos activado (led verde encendido), que de todos modos es mas rápido que la velocidad del tren original de Lego Duplo.
- **NOTA SOBRE LOS SONIDOS (PITIDOS) Y LUCES:**
- Normalmente cuando **seleccionamos** una opción se producen **2 pitidos** y cuando la **deseleccionamos** se produce un **sonido "apagado"**. Podéis ver el dibujo del mando a distancia para saber que hace cada tecla.
- Cuando **se llega a la velocidad máxima** (tiene que estar el control de velocidad por ultrasonidos deshabilitado (**led verde apagado**) y subimos la velocidad pulsando varias veces el pulsador de avance (si estamos avanzando) o retroceso (si estamos marcha atrás), al llegar a la velocidad máxima **se produce 1 pitido**. También sucede lo mismo cuando le damos al motor porta imanes varias veces en el mismo sentido de giro, (**1 pitido = llegó a la velocidad máxima de giro**).
- Cuando pulsamos la tecla **"100+"** de nuestro mando se produce el sonido llamado "Claxon" que son **3 pitidos**.
- Podemos deshabilitar todos los "sonidos" (pitidos) si pulsamos la tecla número **"200+"** de nuestro mando infrarrojo (VS1838), **recordarlo "padres"** para tener un poco de "paz" y nunca comentárselo a vuestros hijos..... ☺☺☺
- **ACTUALIZACION:** También se ha añadido que cuando se habilite o deshabilite alguna opción, las luces delanteras nos hagan un par de parpadeos para señalizar que se ha activado/deshabilitado una opción (esto siempre lo hace incluso con los sonidos desactivados). También en la versión del programa **"v1.2"** podemos bajar la velocidad gradualmente hasta llegar a parar cada motor (dando varias pulsaciones) para los 2 motores.
- También se ha añadido el poder controlar 2 trenos con un mismo mando (**se tiene que dejar teclas comunes para ambos**) ya que no hay teclas suficientes para dejar todas las opciones programadas individualmente. **Simplemente tendremos 2 programas y cargaremos uno diferente a cada coche**. Nombre de archivo **"Tren_Lego_Duplo_Sofia_v1.2_Tren1.ino"** y **"Tren_Lego_Duplo_Sofia_v1.2_Tren2.ino"**

- ALGUNAS FOTOS







- Y esto es todo, si tenéis alguna duda o pregunta de cómo realizar este proyecto, dejar un comentario en la página de Thingiverse. Un Saludo.