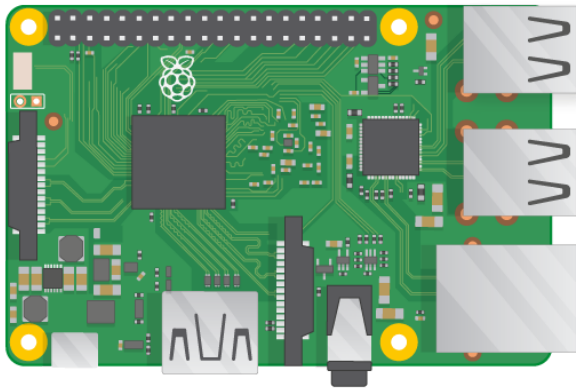


Telemetría y Sensórica del prototipo de Trazabilidad

A continuación se explica el funcionamiento general del Hardware del prototipo de Trazabilidad, con el fin de explicar la recolección y el envío de información y su posterior implementación en cualquier Raspberry que cuente con los módulos necesarios para su funcionamiento. Esto basado en los ejecutables creados que se encuentran en el Github creado para este fin. Información de los módulos usados y de la Raspberry, se puede encontrar en el apartado de **Documentation** encontrada en el Github.

Raspberry Pi3 - Screen 3.5' Resistivo



La Raspberry Pi3 es un sistema embebido que necesita una Micro SD para su funcionamiento, ya que es en este almacenamiento que la Raspberry puede arrancar el sistema operativo, por motivos de simplicidad, se escogió el SO que viene por defecto en la página oficial de Raspberry. Este debía ser modificado para que la pantalla funcionará correctamente desde que la Raspberry sea arrancada.

Con ayuda de la página oficial de ADAFRUIT, compañía distribuidora del Screen, se usó unos pasos que ellos dan para poder utilizar la pantalla junto con la Raspberry, en la página oficial de ADAFRUIT se puede encontrar los pasos necesarios. A continuación se muestra donde se descargó la última versión de Sistema Operativo de la Raspberry

Imagen Jessie - Lasterd

Con la imagen, se copio la información en la microSD, según los pasos de la página oficial de Raspberry, los cuales se pueden replicar y están descritos para cualquier Sistema Operativo. Esto se encuentra en el siguiente link en la sección denominada **WRITING AN IMAGE TO THE SD CARD**.

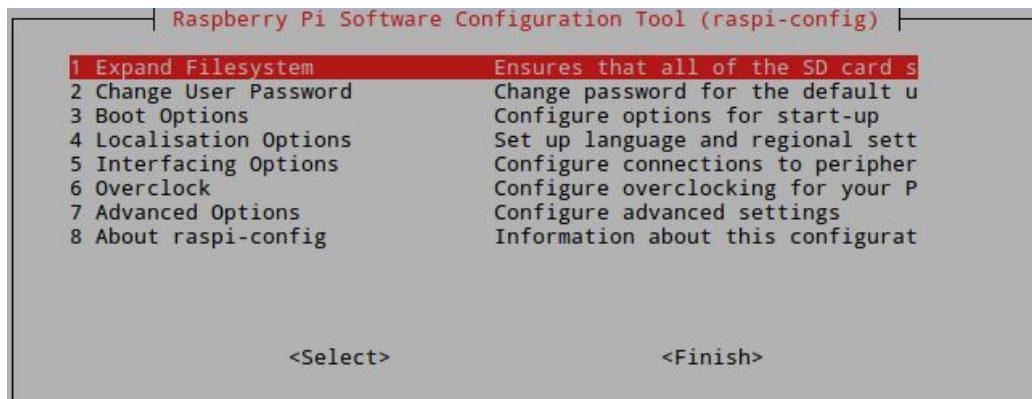
Pasos Instalación Imágen

Con la imagen instalada en la MicroSD, se conectó la Raspberry y se siguieron los siguientes pasos para poder inicializar correctamente la Raspberry para su funcionamiento con los programas creados y la pantalla. Los siguientes comandos son hechos desde la *Terminal* de la Raspberry:

```
$ sudo raspi-config
```

Se verá una ventana como se muestra a continuación, en la sección de ExpandFile es usada para utilizar todo el espacio posible de la MicroSD ya que al inicio solo usa 4GB de

las posibles 32GB de las MicroSD compradas. Después en la sección 4 *Interfacing Options*, se habilita la opción de comunicación serial, usada para uno de los módulos del proyecto. Después al darle la opción de *<Finish>* el sistema pedirá un reinicio, este será necesario para poder tener los cambios realizados.



```
$ sudo apt-get update && sudo apt-get upgrade

$ curl -SLs https://apt.adafruit.com/add-pin | sudo bash

$ sudo apt-get install -y raspberrypi-bootloader
adafruit-pitft-helper raspberrypi-kernel

$ sudo adafruit-pitft-helper -t 35r

$ sudo nano /boot/config.txt
```

Con la primera de las líneas el proceso será lento debido a que actualizará el sistema operativo de la Raspberry a su última versión, lo cual al ser primera vez demorará unos minutos dependiendo de la conexión de internet que se tenga en estos momentos. Las siguientes son para instalar el Screen de 3.5' y con la última se accede a un archivo encargado de ejecutar el sistema operativo en la Raspberry al estar conectada. Para poder acceder exitosamente a uno de los módulos es necesario usar un protocolo de comunicación llamado UART. La Raspberry dentro de su Hardware cuenta con dos de estos protocolos, sin embargo, el segundo funciona a una velocidad mucha menor al principal y ocasiona conflictos cuando se intentó usar. Por este motivo es necesario usar el UART principal de la Raspberry el cual en su última versión (RBPi3) se encuentra usado para la conexión Bluetooth del dispositivo. Por este motivo es necesario deshabilitar esta opción para poder acceder a este protocolo. Por lo cual al final del archivo **config.txt** se le agregó el siguiente comando:

```
dtoverlay=pi3-disable-bt
```

Esto hará que el Bluetooth sea deshabilitado para poder llevar el puerto UART principal a los GPIO de la Raspberry. Nuevamente se reinicia la Raspberry, esto se puede hacer desde consola con el siguiente comando:

```
$ sudo reboot
```

Finalmente accedemos al archivo **cmdline.txt** para eliminar la parte de este que dice **console=serial0,115200**, una vez eliminado esta parte del archivo nuevamente ejecutamos el comando de reinicio. Finalmente, la Raspberry ya cuenta con el UART principal en el GPIO para que se pueda usar correctamente.

```
$ sudo nano /boot/cmdline.txt
```

FONA 808 - Módulo GPS y GSM



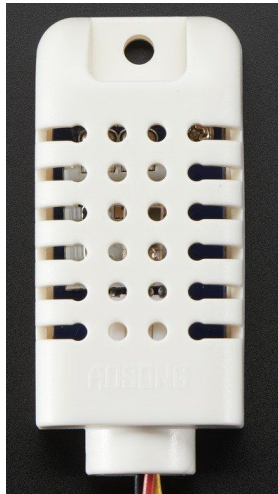
Este módulo es el encargado mediante las antenas GPS y GSM compradas de recibir la información GPS de los satélites (Hora, Latitud, Longitud y Altitud) y ser enviada después del proceso de validación hecho mediante el servidor local de la Raspberry.

Como se puede ver, en primera medida se tiene que soldar la regleta para poder acceder a los pines necesarios para la alimentación y el envío de información al módulo. Al usarlo de esta manera podemos conectar a la protoboard para poder acceder fácilmente a los pines y poderlos conectar a la Raspberry.

Debido a las limitaciones de la alimentación del módulo, este dispositivo trabaja con antenas pasivas, es decir, antenas que no tienen una amplificación anterior que amplifique la ganancia de la señal captada. Sin embargo, debido a que se necesitaba que la antena tuviera buena recepción en cualquier zona, se decidió usar antenas activas para el caso del GPS y de esta forma poder captar la señal mucho más rápido. De esta forma es necesario entonces usar una alimentación además del módulo a la antena para que esta pueda funcionar.

Finalmente, para que el todo funcione correctamente es necesario primero cargar el ejecutable llamado **pwrSIM.py** y encender el dispositivo con el número **1**. Después de esto se puede recibir la información *GPS (Fecha, Latitud, Longitud y Altitud)* y enviar cualquier información (*POST*) o recibir información (*GET*), con los otros archivos python ejecutables que se encuentran en el repositorio.

AM2302 - Sensor de Temperatura y Humedad



El sensor de temperatura y humedad es un dispositivo que mediante un protocolo llamado *One-Wire*, envía la información de temperatura y humedad del ambiente. Para esto solo necesita los cables de la alimentación y el cable de envío de información. Ayudados de la librería que da el fabricante para sus dispositivos se modificó esto para poder utilizar este módulo en específico. Para poder utilizar el ejecutable **getTemHum.py** es necesario primero ejecutar los siguientes comandos:

```
$ sudo apt-get install build-essential python-dev  
$ sudo python setup.py install
```

Este último comando se hace estando en la carpeta del repositorio que se encuentra en /Script/Sensor_Library/, así queda instalado la librería del sensor y no se necesita importar el paquete.

SL018 - Lector RFID



Este lector será usado con el objetivo de reconocer a los productores y transportadores de la cadena, en general se espera que cada actor pueda ser reconocido mediante el lector RFID, mediante la información presente en el servidor local. Para su funcionamiento es necesario únicamente conectar la Raspberry correctamente ya que el sistema está elaborado para que el programa se ejecuta internamente recién se prende el sistema operativo de la Raspberry. Al igual que el módulo FONA 808 es necesario adecuar una regleta a las cinco huecos que se ven en la parte superior para poder acceder a la alimentación y reconocimiento del TAG (tarjeta, adhesiva, llavero). La distancia de detección es de aproximadamente cinco centímetros del módulo para que este reconozca el número correctamente.

PINOUT General del Prototipo

A continuación se presenta las conexiones que son necesarias para poder utilizar los códigos del repositorio sin problema alguno. Si bien esto está centrado en la Raspberry, utilizando la pantalla 3.5' las conexiones son iguales debido a que los pines que utiliza la pantalla los vuelve a dejar libres en las otras conexiones que tiene la misma, de esta manera sería replicar estas conexiones en la pantalla cuando está se quiera conectar.

