**Multicell Battery Monitor Based on LTC6802G-1 Chip**

**Test SPI Commands and Data Flow from Master Arduino to LTC6802 Chip**

# **Introduction**

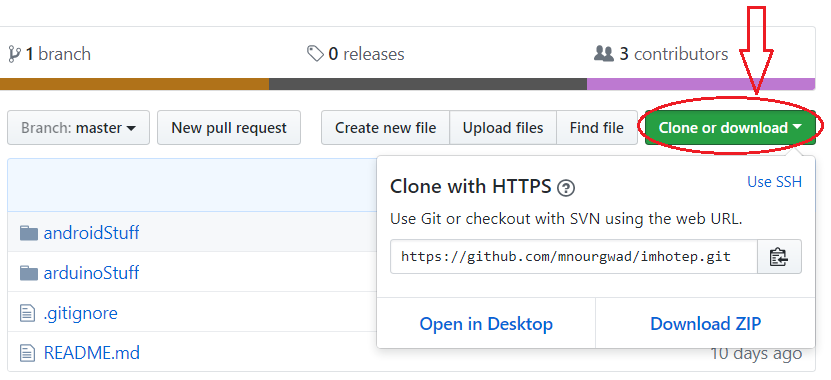
This document provides the steps needed to use the slave-side simulation in SPI communication between Arduino chips. This is mainly written to be able to test the commands and data flow from the master Arduino chip to the chip LTC6802. This simulation simply printout any data received from the master so that one can check the communication errors and debug it.

# **User Manual:**

## Step (1): Getting things on PC.

Clone or download the imhotep repository from the following link:

<https://github.com/mnourgwad/imhotep/>



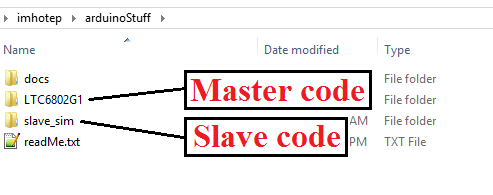
Fig(2):

## Step (2): Reaching the code.

Now you have ‘imhotep’ folder on you PC. It contains two main subfolders; ‘**androidStaff**’ and ‘**arduinoStaff**’. In this tutorial we work on the Arduino side.

In arduinoStaff subfolder we have two Arduino projects:

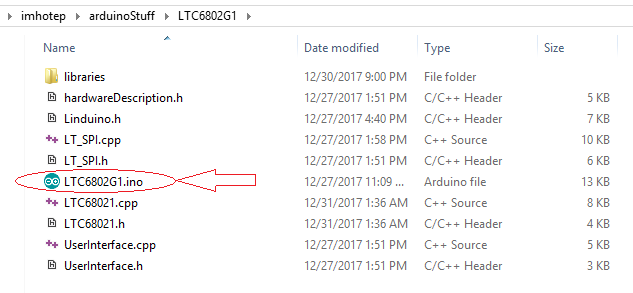
1. ‘**LTC6802G1**’: This implements the master side.
2. ‘**slave-sim**’: This implements a slave-side in the SPI communication to simulate the IC operation and test the proper instruction and data flow controlled by the master side.



Fig(3):

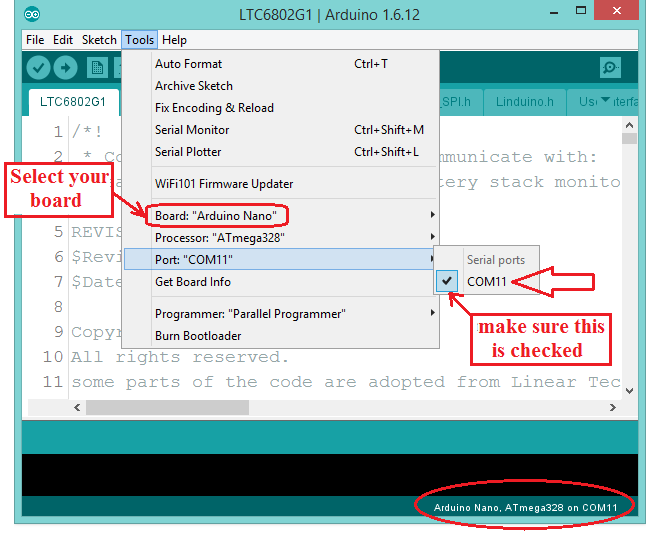
## Step (3): Running the master-side.

1. Open ‘LTC6802G1’ folder.
2. Open ‘LTC6802G1.ino’ file. It will open in the Arduino IDE by default.



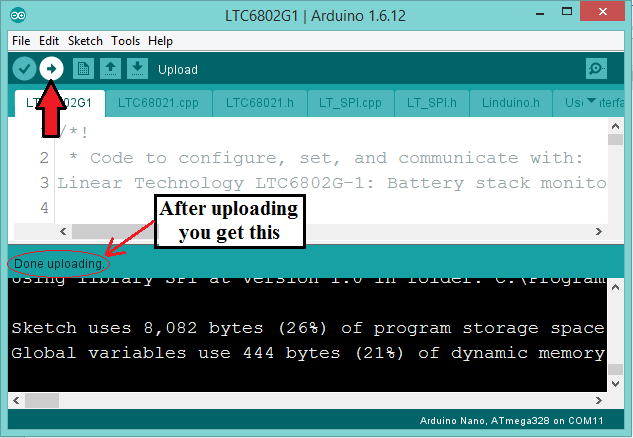
Fig(4):

1. Connect your Arduino unit ‘Master’ to any serial port of your computer. Make sure that you selected the right Arduino board and serial port. In our tutorial we use Arduino Nano. Here is what we must get the following:



Fig(5):

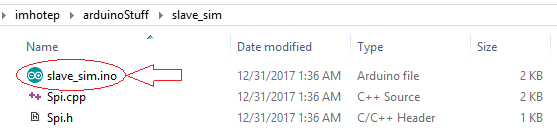
1. Now we can upload the code on our board by clicking the ‘upload’ icon on the top-left corner of the Arduino window.



Fig(6):

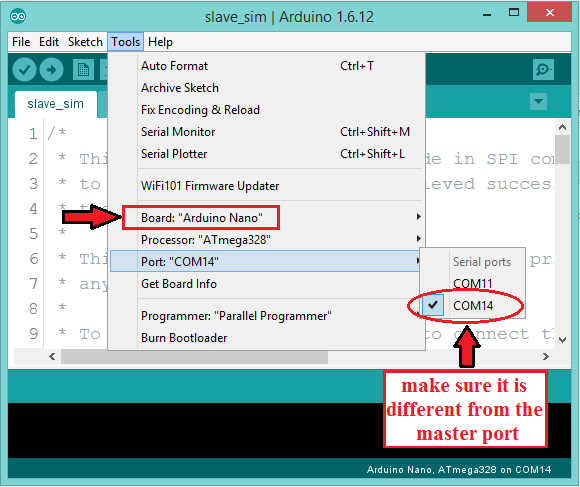
## Step (4): Running the salve-side.

1. Open ‘slave\_sim’ folder.
2. Double click on ‘slave\_sim.ino’ file. It will open in the Arduino IDE by default.



Fig(7):

1. Connect your Arduino unit ‘Master’ to any serial port of your computer. Make sure that you selected the right Arduino board and serial port. In our tutorial we use Arduino Nano. Here is what we must get the following:



Fig(8):

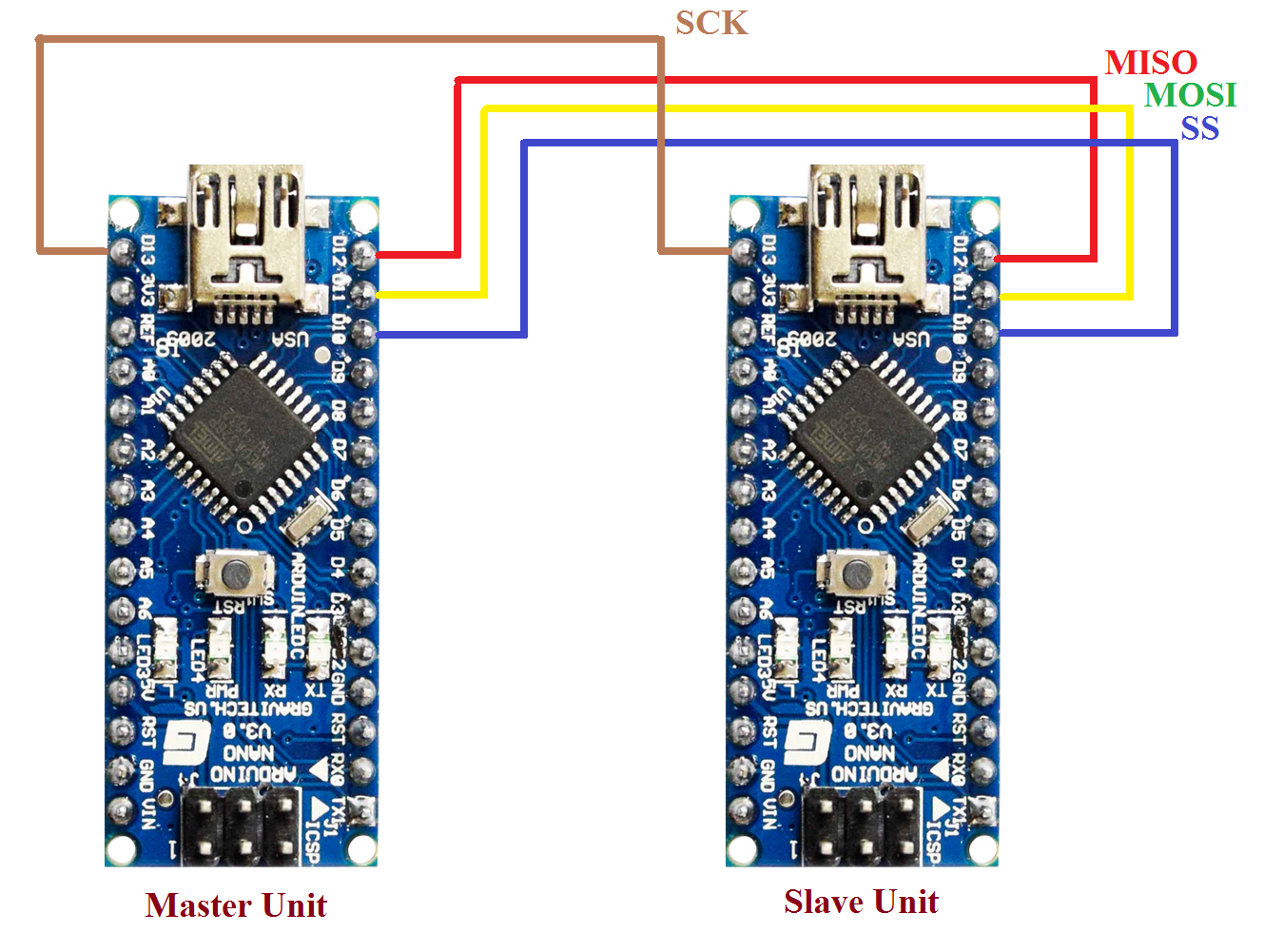
1. Click on the upload icon to upload the code on the slave board.

## Step (5): Connecting slave to master (Hardware wiring).

In Serial Peripheral Interface (SPI) communication there are 4 pins to connect. The following lines describes these pins and their numbers in Arduino **Nano** board.

1. Slave Select (SS) pin 10
2. Master Out Slave In (MOSI) pin 11
3. Master In Slave Out (MISO) pin 12
4. Serial clock (SCK) pin 13

Connect each pin in master with the same pin in slave. Your circuit must be something like this:

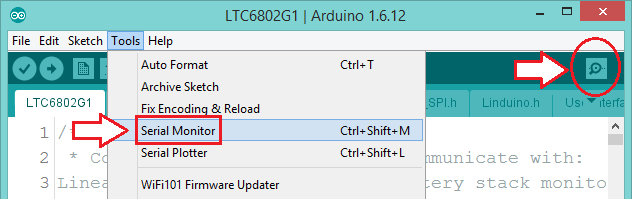


Fig(9):

## Step (6): Running the Serial Monitor.

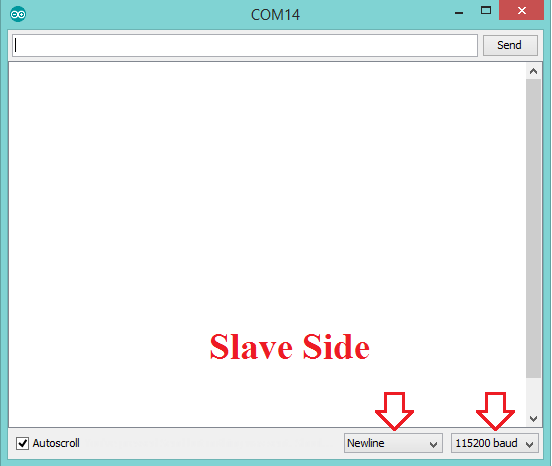
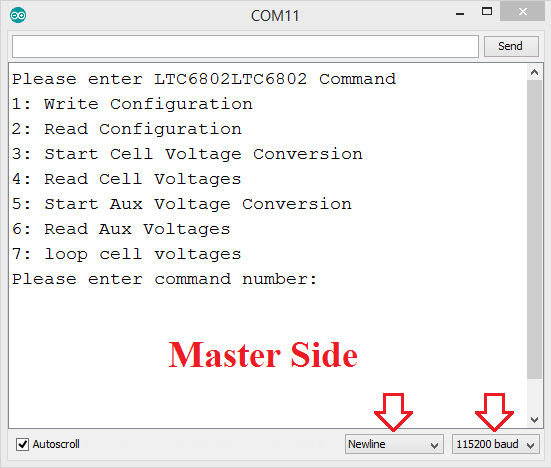
There are two ways to open the serial monitor in Arduino IDE:

1. Click the icon in the top right corner.
2. From ‘Tools’ choose ‘Serial Monitor’ option.



Fig(10):

Open the serial monitor in both master and slave Arduino windows. Make sure that the selected baud rate in the bottom right corner in the serial monitor is 115200 as it was initialized in the code.

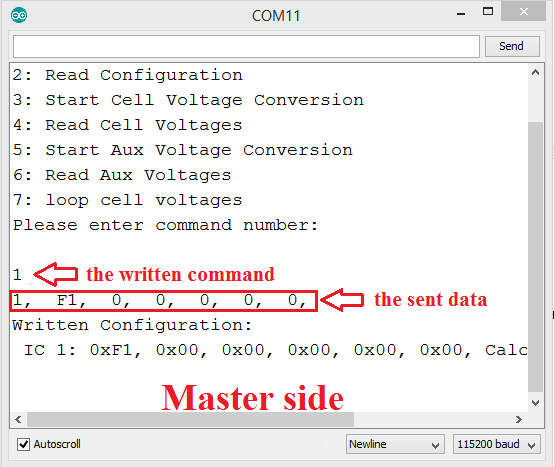
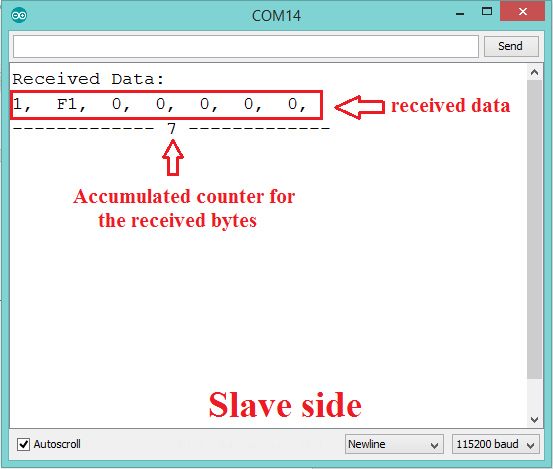


1. (b)

Fig(11):

## Step (7): Applying the Test.

In the serial monitor of master side you can write a command number from 1 to 7 and check the received data in the serial monitor of slave side. Here is what we get when we write command (1) and press ENTER key

1. (b)

Fig(12):