DESCRIPTION DOCUMENT FOR LoRaWAN BOARD

HARDWARE REVISION 0.3

Department	Name	Signature	Date
Author			
Reviewer			
Approver			

Revision History

Rev	Description of Change	Effective Date
A	Initial Release	

ABSTRACT:

This document is a detailed product description that describes the effective features of the product. It includes a functional hardware description of the product with its internal block diagram and product images.

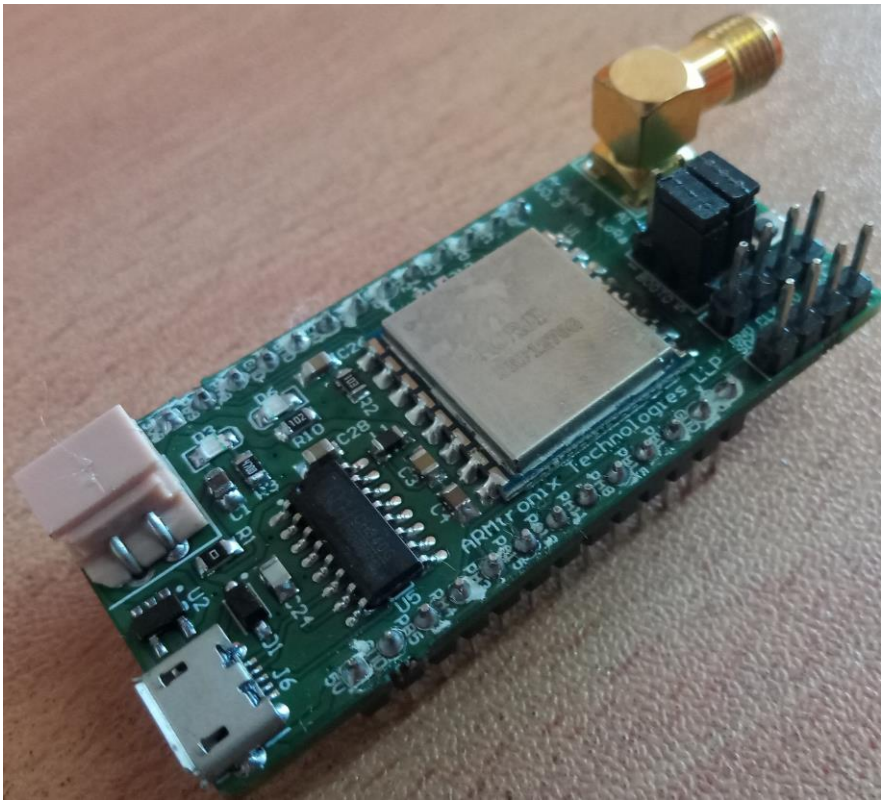

Table of Contents

Table of figures

1. ABBREVIATIONS

Term	Description
CPU	Central Processing Unit
DC	Direct Current
GPIO	General Purpose Input Output
I2C	Inter Integrated Circuit
IDE	Integrated Development Environment
Li-Ion	Lithium Ion
MCU	Microcontroller Unit
mm	Mili Meter
PCB	Printed Circuit Board
SCL	Serial Clock
SDA	Serial Data
STM	ST Microelectronics
UART	Universal Asynchronous Receiver Transmitter
USART	Universal Synchronous Asynchronous Receiver Transmitter
USB	Universal Serial Bus
V	Voltage
WAN	Wide Area Network

2. REFERENCES

Company Website link	https://www.armtronix.in
Intractable's Weblink	
Github's Weblink	https://github.com/armtronix/arduino-LoRa-STM32

3. PURPOSE

The purpose of this document is to outline the design description for the LoRa WAN with STM controller evaluation Board. It provides a high level summary of the product.

4. SCOPE

This document describes system architecture which includes Power supply, Microcontroller and Lora Module.

5. SAFETY AND WARNING

If you are working with Li-ion battery, please take necessary precautions. Do not short the positive and negative terminals of the battery, as it may damage the battery and may create hazardous to your health. Please consider disconnecting battery or power supply from the board if you would like to make any changes in connections. Working without safety towards Li-ion batteries is not advisable.

6. INTRODUCTION

LoRa node is a LoRa technology based device can be used for remote monitoring and controlling application where a distance is a constraint. This board has STM microcontroller with DRF176G LoRa module interfaced through SPI communication and the board is designed to support for LoRa WAN protocol. The STM microcontroller used is this board is Arduino IDE compatible, so that users need not to worry about external programmer. This board has the on board single cell lithium ion battery charger for battery backed application. It also has dedicated I2C header and most of the GPIOs of controller are accessible to user for their specific application.

7. PRODUCT FEATURES

- Works directly with DC input 4.5 V – 5.5 V.
- Product firmware can be updated/reloaded/changed as per user requirement.
- It has onboard USB to UART converter on board for programming purpose.
- Board is Arduino compatible.
- Has a LoRa moudule and board is LoRaWAN compatible.
- Basic Firmware to enter SSID and password to connect to the router
- Has most of all the GPIOs of controller are accessible for user's application.
- Push Button on board Provided to Restart.
- Single cell lithium-ion battery charger available on-board.

8. PRODUCT DESCRIPTION

a. PHYSICAL DESCRIPTION

- DC to DC Power supply module
- LoRa Module.
- Li-Ion Charger (Single Cell)
- Microcontroller

b. FUNCTIONAL DESCRIPTION

Block Diagram

LoRa node has an on-board DC-DC voltage regulator which takes DC input of 5V and provides regulated DC power as output for both controller and LoRa module. The same input voltage and regulated voltages are available in the headers provided for user access so that user can make use of it to power up low power consumption sensor.

9. SYSTEM OVERVIEW

1. DC to DC Power supply module

The DC-DC converter on board is used to regulate voltage from 5 V DC to 3.3 V DC to supply power to complete digital part.

2. LoRa Module

LoRa module DRF1276G used on board for LoRa commutation.

3. Microcontroller

The Microcontroller executes the application algorithm to transmit and receive the data to and from LoRa network. The STM32F103CBT6 used on the board, has sufficient memory to load LoRa stack.

10. TECHNICAL SPECIFICATION

a. ELECTRICAL SPECIFICATION

Input Specifications				
Description	**Min**	**Typ**	**Max**	**Unit**
Voltage DC	4.5	5	5.5	Volts
Current DC	-	0.5	-	Amps

b. MECHANICAL SPECIFICATION

- Mechanical Dimensions of PCB are 55 x 27 X 15mm (Length x Width x Height)

11. ELECTRICAL CONNECTIONS

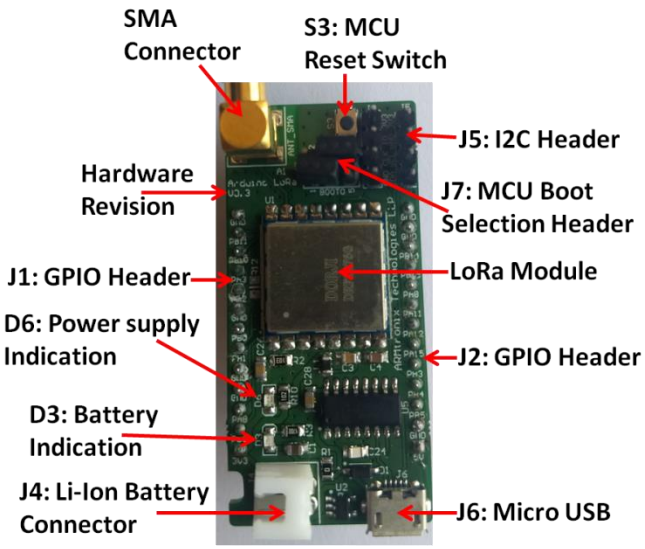

Figure 1: Header and Switch Details

Description of Header and Switches shown in Figure 1:

1. J1: MCU GPIO Header
2. J2: MCU GPIO Header
3. J4: Li-ion Battery connector
4. J5: I2C Header
5. J6: Micro USB connector for programming
6. J7: MCU Boot selection Header

a. Header J1:

Header Pin No.	MCU Pin No.	Pin Name
1	-	VCC_3V3
2	46	PB9/CANTX
3	45	PB8/CANRX
4	-	DGND
5	-	DGND
6	19	PB1
7	18	PB0
8	-	DGND
9	12	PA2/USART2_TX
10	13	PA3/USART2_RX
11	21	PB10
12	22	PB11
13	-	DGND

Table 1: Header J1 Pin configuration

b. Header J2:

Header Pin No.	MCU Pin No.	Pin Name
1	-	VCC_5V
2	-	DGND
3	41	PB5
4	40	PB4
5	39	PB3
6	38	PA15
7	33	PA12
8	32	PA11
9	29	PA8
10	28	PB15
11	27	PB14
12	-	DGND
13	-	DGND

Table 2: Header J2 Pin configuration

c. Header J4:

Header Pin No.	Battery Pin No.
1	Positive (+)
2	Negative (-)

Table 3: Header J4 Pin configuration

d. Header J5:

Header Pin No.	MCU Pin No.	Pin Name
1	-	VCC_3V3
2	-	DGND
3	42	I2C_SCL
4	43	I2C_SDA

Table 4: Header J5 Pin configuration

e. Header J7:

Header Pin No.	MCU Pin No.	Pin Name
1	-	VCC_3V3
2	-	VCC_3V3
3	44	BOOT0
4	20	BOOT1
5	-	DGND
6	-	DGND

Table 5: Header J7 Pin configuration

f. Hardware connections between MCU and LoRa

LoRa Pin No.	LoRa Pin Name	MCU Pin No.	MCU Pin Name
1	Reset	2	PC13
2	DIO0	11	PA1
3	DIO1	26	PB13
4	DIO2	25	PB12
9	SCK	15	SCK
10	MISO	16	MISO
11	MOSI	17	MOSI
12	NSS	14	SS

12. HOW TO CUSTOMISE FIRMWARE

You can program this board using Arduino IDE. Please follow the below steps to program the board by yourself with easy steps as mentioned below:

1. Install Arduino board manager related for STM32F103CBT6.
2. Open your code in Arduino IDE.
3. Click on Tools Tab, move mouse pointer on “Board: “xxxxxxxx” and click on “Generic STM32F103C Series” as shown in figure 2.

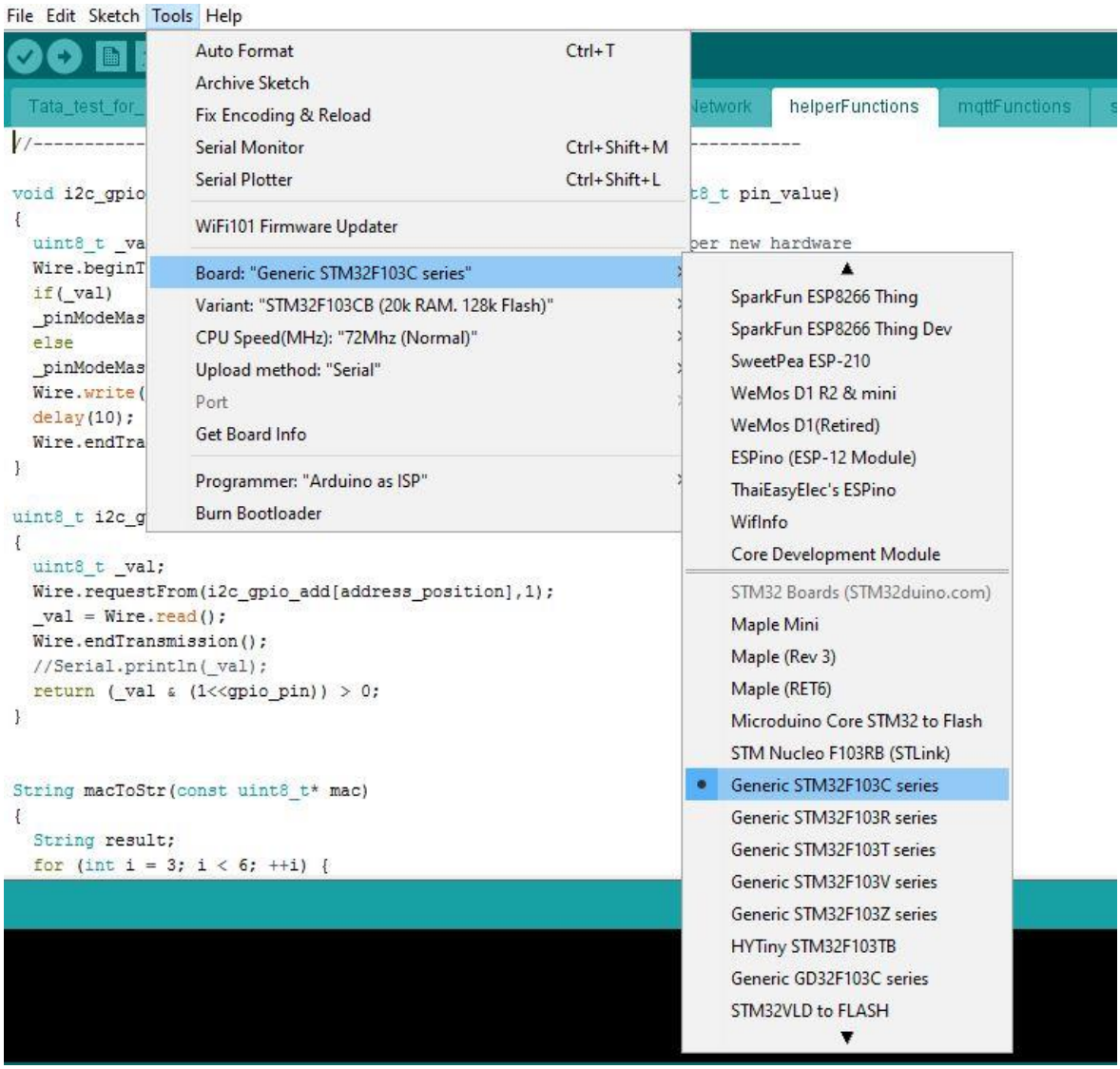

Figure 2: Board Selection

4. Click on tools tab, move mouse pointer to “Variant: “xxxxx”, under this click on “STM32F103CB” to select. Refer to figure 3.

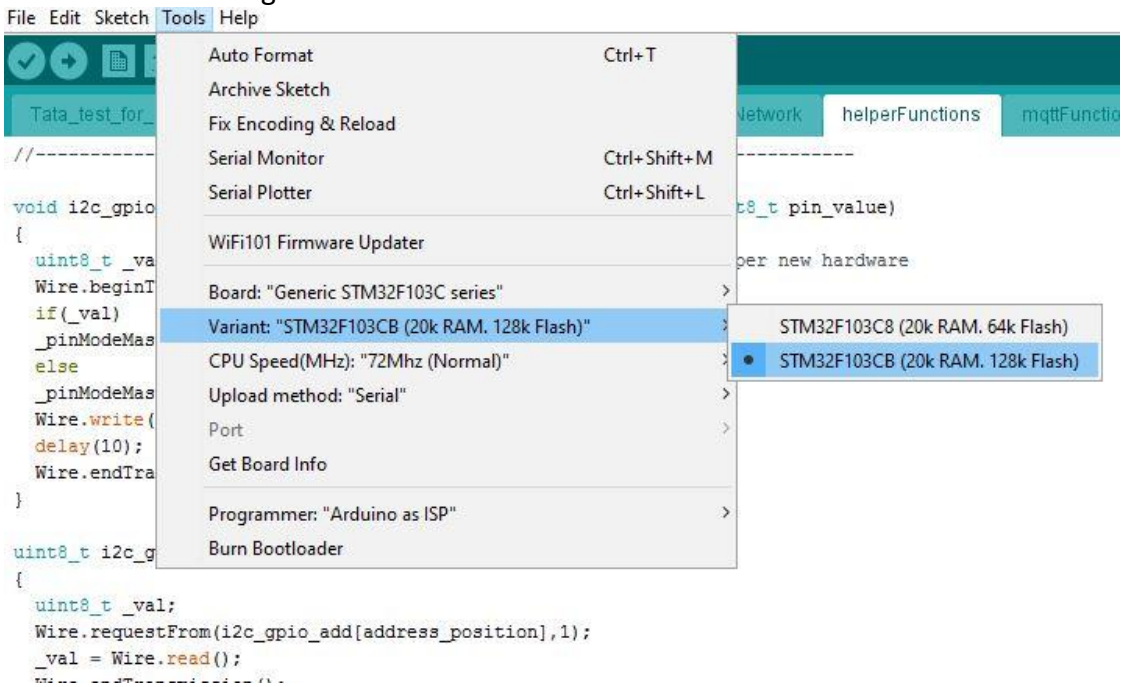

Figure 3: Variant Selection

5. Click on tools tab, move mouse pointer to “CPU Speed(Mhz): xxxx”, under this click on “72Mhz (Normal)” to select. Refer to figure 4

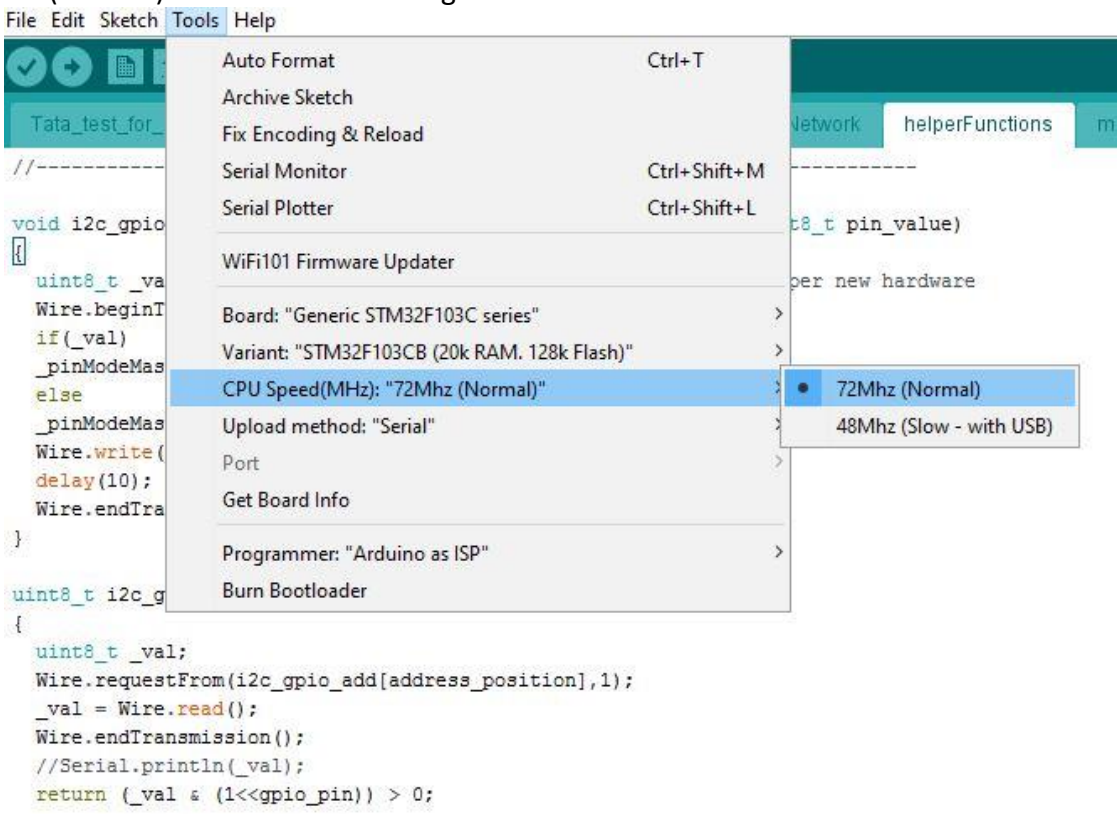

Figure 4: CPU Speed selection

6. Click on tools tab, move mouse pointer to “Upload Method: “xxxxx”, under this click on “Serial” to select. Refer to figure 5.

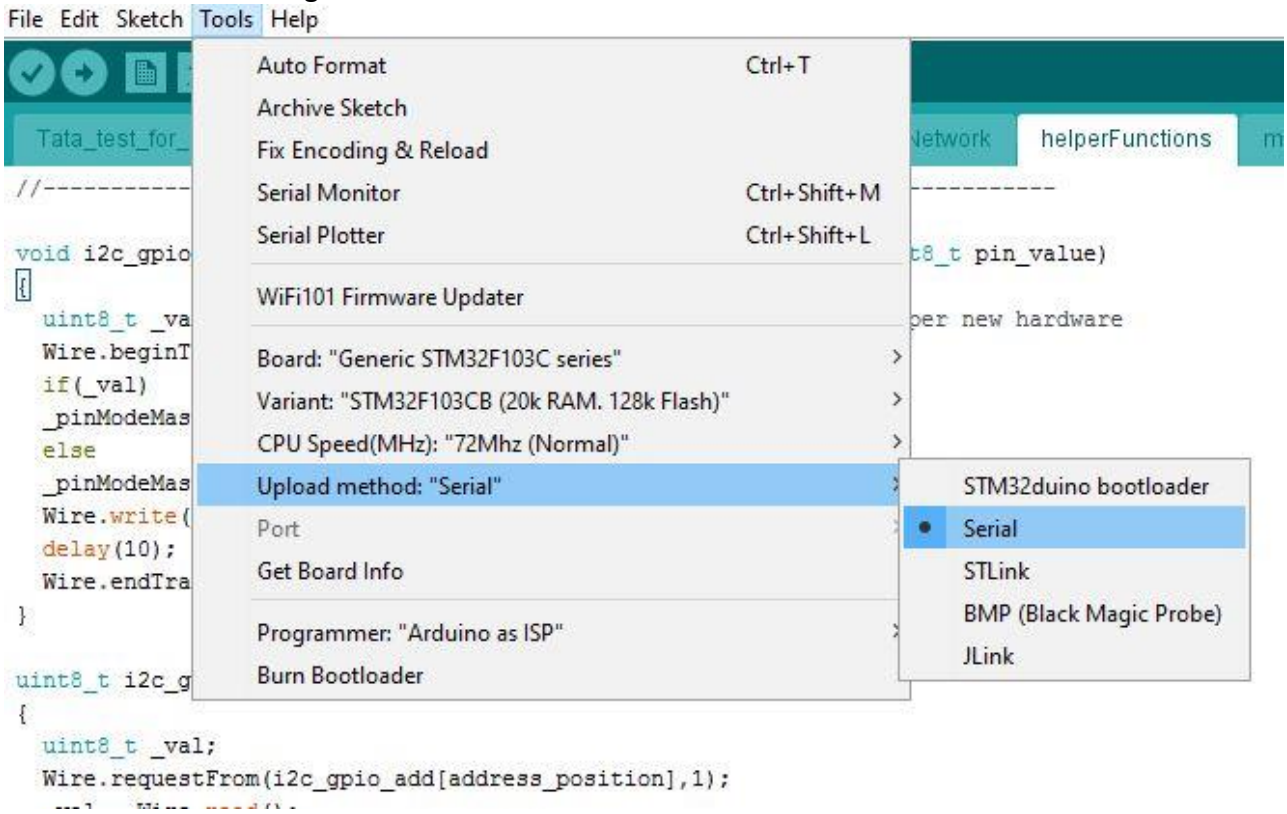

Figure 5: Upload method selection

7. Click on tools tab, move mouse pointer to “Port: “COMx”, under this click on “COMx” to select. (“x” refers to port number available in your computer) Refer to figure 6.

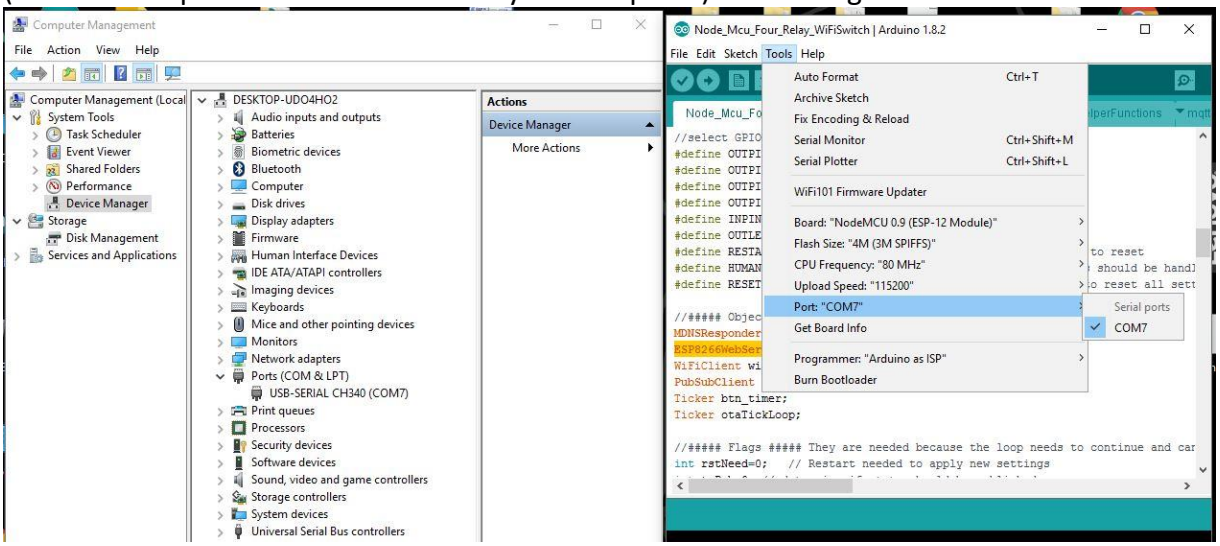

Figure 6: COM port selection.

8. Click on tools tab, move mouse pointer to “Programmer: “Arduino as ISP’”, under this click on “Arduino as ISP” to select. Refer to figure 7.

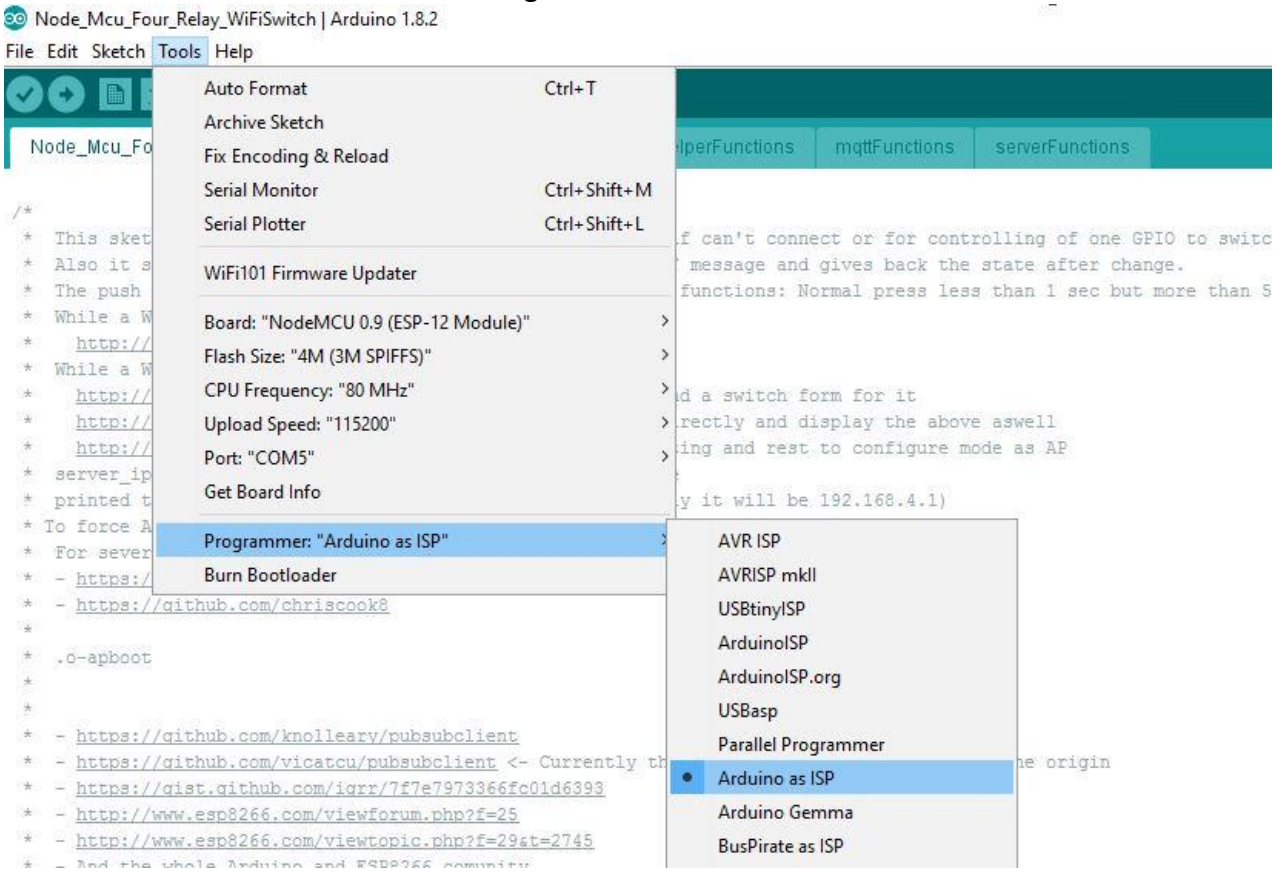

Figure 7: IDE Selection

9. Run the program. Refer to Figure 8.

Node_Mcu_Four_Relay_WiFiSwitch | Arduino 1.8.2

File Edit Sketch Tools Help

Upload

Node_Mcu_Four_Relay_WiFiSwitch | ConfigFunctions | Network | helperFunctions | mqttFunctions | serverFunctions

```
//select GPIO's
#define OUTPIN_04 4 //output pin
#define OUTPIN_12 12 //output pin
#define OUTPIN_13 13//output pin
#define OUTPIN_14 14 //output pin
#define INPIN 0  // input pin (push button)
#define OUTLED 5
#define RESTARTDELAY 3 //minimal time in sec for button press to reset
#define HUMANPRESSDELAY 50 // the delay in ms untill the press should be handled as a normal push by human.
#define RESETDELAY 20 //Minimal time in sec for button press to reset all settings and boot to config mode

//##### Object instances #####
MDNSResponder mdns;
ESP8266WebServer server(80);
WiFiClient wifiClient;
PubSubClient mqttClient;
Ticker btn_timer;
Ticker otaTickLoop;

//##### Flags ##### They are needed because the loop needs to continue and cant wait for long tasks!
int rstNeed=0;   // Restart needed to apply new settings
int toPub=0; // determine if state should be published.
int configToClear=0; // determine if config should be cleared.
int otaFlag=0;
boolean inApMode=0;
//##### Global vars #####
int webtypeGlob;
int otaCount=300; //imeout in sec for OTA mode
int current; //Current state of the button
```

Figure 8: Running code

IMPORTANT NOTICE

ARMtronix Technologies LLP and its subsidiaries reserve the right to make corrections, enhancements, improvements and other changes to its products and services and to discontinue any product or service. Buyers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to ARMtronix Technologies LLPs terms and conditions of sale supplied at the time of order acknowledgment.

The information in this document is subject to update without notice. The contents of this document thereof must not be used for any unauthorized purpose.

----------END OF DOCUMENT----------