Grove Analog Moisture Sensor on BeagleBone Guide

By Group Name

Introduction

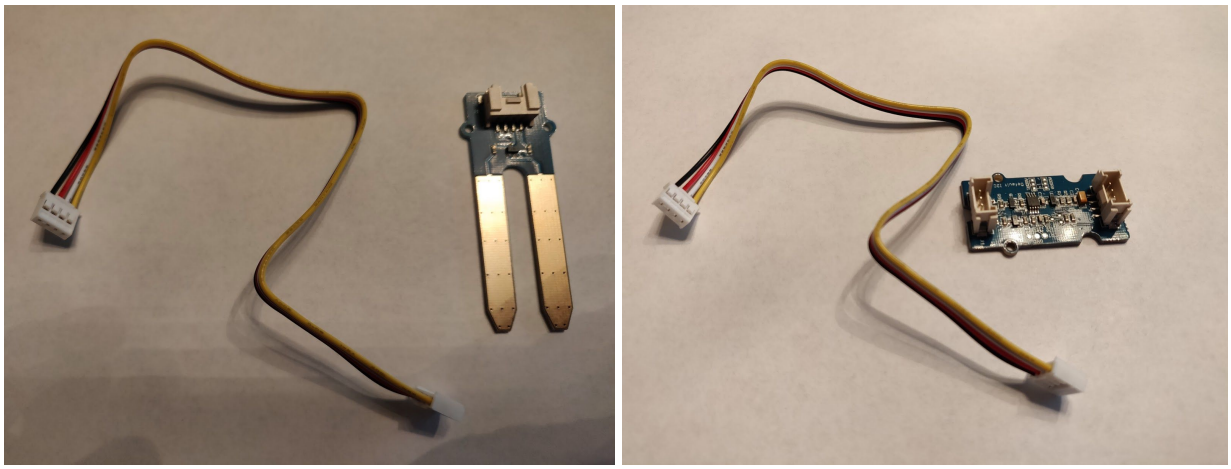
A Grove moisture sensor is used to determine the amount of moisture present around the sensor. It is most commonly used to measure the moisture content of soil. According to the seeedstudio wiki, it is not compatible with the BeagleBone. However, this guide will cover how to interface the analog sensor to the digital UART port on the BeagleBone Green through both the command line and C code.

Table of Contents

1. Required Parts

The following parts are required in order to connect the moisture sensor to the grove ports on the BeagleBone Green:

- 1 Grove - Moisture Sensor [1]
- 1 Grove - I2C ADC [2]
- 2 Grove Cables



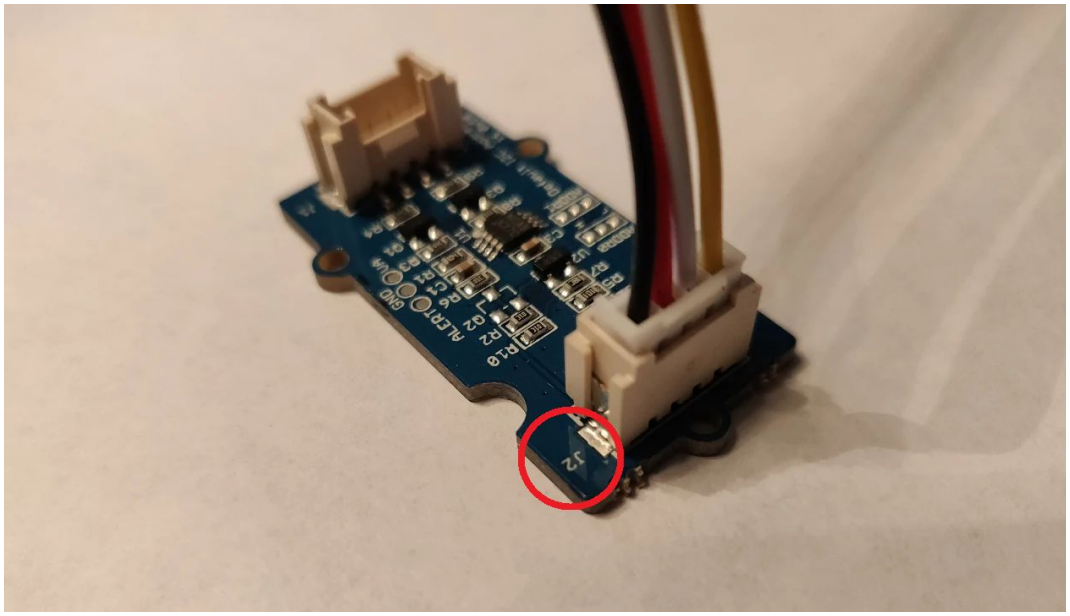
Grove - Moisture Sensor (Left), Grove - I2C ADC (Right)

[1] http://wiki.seeedstudio.com/Grove-Moisture_Sensor/

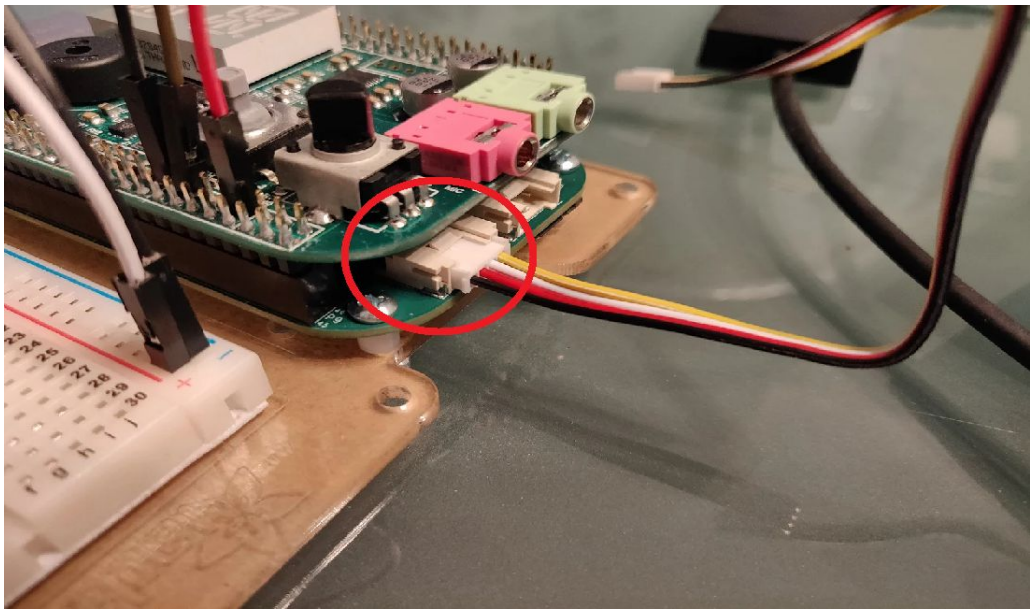
[2] http://wiki.seeedstudio.com/Grove-I2C_ADC/

2. Grove Setup

To connect the Moisture Sensor to the I2C ADC, take a Grove Cable and connect one end into the Moisture Sensor. Take the other end and connect it into the input socket (J2) port of the I2C ADC. The socket labels are next to the Grove sockets.



Connect the second Grove Cable and connect it into the output socket (J1) on the I2C ADC. Plug the other end of the Grove Cable into the I2C Grove Interface on the BeagleBone Green. Be sure not to insert the Grove Cable into the UART Grove Interface.



3. Checking Sensor Output via Command Line

This step assumes that the univ-emmc virtual cape is disabled on the BeagleBone Green. If it is not, then refer to Brian Fraser's I2C Guide: 14-Seg Display, section 2.1. In addition, this step assumes that the I2C tools is installed. If not, then follow step 1 in Brian Fraser's I2C Guide: 14-Seg Display, section 2.2.

1. Ensure that the I2C-2 bus is enabled:

   ```
   # i2cdetect -l
   i2c-0 i2c   OMAP I2C adapter  I2C adapter
   i2c-2 i2c   OMAP I2C adapter  I2C adapter
   ```

   If the i2c-2 bus does not appear, then load the I2C-2 virtual cape:
   `# echo BB-I2C2 > /sys/devices/platform/bone_capemgr/slots`

2. Confirm that the Grove I2C ADC is correctly connected by displaying the I2C devices on the I2C-2 bus:

   `# i2cdetect -y -r 2`

   ```
   root@ethanjsfu:~# i2cdetect -y -r 2
        0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
   00:          -- -- -- -- -- -- -- -- -- -- -- -- --
   10: -- -- -- -- -- -- -- -- UU -- -- -- -- -- -- --
   20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
   30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
   40: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
   50: 50 -- -- -- UU UU UU UU -- -- -- -- -- -- -- --
   60: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
   70: -- -- -- -- -- -- -- --
   ```

   No device found: "--"
   Device at address "##" found: "##"
   In use by driver: "UU"

3. To display the internal memory of the I2C ADC, use the command:
   # i2cdump -y 2 0x50

```
root@ethanjsfu:~# i2cdump -y 2 0x50
No size specified (using byte-data access)
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f    0123456789abcdef
00: 02 00 00 00 0f 00 00 02 00 00 00 00 00 00 00 00    ?...?..?........
10: 32 00 00 00 ff 00 00 32 00 00 00 00 00 00 00 00    2......2........
20: 02 00 00 00 0f 00 00 02 00 00 00 00 00 00 00 00    ?...?..?........
30: 32 00 00 00 ff 00 00 34 00 00 00 00 00 00 00 00    2......4........
40: 02 00 00 00 0f 00 00 02 00 00 00 00 00 00 00 00    ?...?..?........
50: 32 00 00 00 ff 00 00 39 00 00 00 00 00 00 00 00    2......9........
60: 02 00 00 00 0f 00 00 02 00 00 00 00 00 00 00 00    ?...?..?........
70: 32 00 00 00 ff 00 00 39 00 00 00 00 00 00 00 00    2......9........
80: 02 00 00 00 0f 00 00 02 00 00 00 00 00 00 00 00    ?...?..?........
90: 32 00 00 00 ff 00 00 3e 00 00 00 00 00 00 00 00    2......>........
a0: 02 00 00 00 0f 00 00 02 00 00 00 00 00 00 00 00    ?...?..?........
b0: 32 00 00 00 ff 00 00 3e 00 00 00 00 00 00 00 00    2......>........
c0: 02 00 00 00 0f 00 00 02 00 00 00 00 00 00 00 00    ?...?..?........
d0: 32 00 00 00 ff 00 00 3e 00 00 00 00 00 00 00 00    2......>........
e0: 02 00 00 00 0f 00 00 02 00 00 00 00 00 00 00 00    ?...?..?........
f0: 32 00 00 00 ff 00 00 3e 00 00 00 00 00 00 00 00    2......>........
```

   - The I2C ADC stores 12 bytes of data starting at register 0x00.
   - To read the sensor value, take the last 4 bits of register 0x00 as the Most Significant Bits (MSB). Then take the 8 bits of register 0x10 as the Least Significant Bits (LSB).

4. Troubleshooting
   - If there is no device at register 0x50, then ensure that the Grove Cable is connected to the I2C Grove Socket on the BeagleBone Green.
     Ensure that the Moisture Sensor is connected to the J2 socket on the I2C ADC, and that the BeagleBone is connected to the J1 socket.
     The device may also be located on a register from 0x50 to 0x5A if the address on the I2C ADC has been changed from the default address.
   - If you get the output:

     # i2cdetect -l
     `-bash:i2cdetect:command not found`

     The I2C tools for Linux is not downloaded. Follow step 1 in Brian Fraser's I2C Guide: 14-Seg Display, section 2.2.

4. Checking Sensor Output via C Code

4.1 Initialization

The following function initializes the device on I2C-2:

```
#define MOISTURE_SENSOR_ADDR_CONFIG 0x02
#define MOISTURE_SENSOR_ADDR 0x50
#define BB-I2C2 "BB-I2C2"
#define I2C-2 "/dev/i2c-2"
#define SLOTS "/sys/devices/platform/bone_capemgr/slots"

int initializeI2CBus() {
	//Load the I2C2 virtual cape
	FILE *file = fopen(SLOTS, “w”);
	if (file == NULL) {
		printf("Error: could not open file '%s'\n", filename);
	exit(EXIT_FAILURE);
	}

	int res = fprintf(file, “%s”, BB-I2C2);
	if (res <= 0) {
		printf("Error: could not write to file '%s'\n", filename);
		return 1;
	}

	fclose(file);

	//Initialize the I2C-2 bus
	int moistureSensorFD = open(I2C-2, O_RDWR);
	if (moistureSensorFD < 0) {
		printf("Error:  Could not open I2C-2 bus for read/write\n");
		exit(EXIT_FAILURE);
	}

	res = ioctl(moistureSensorFD, I2C_SLAVE, MOISTURE_SENSOR_ADDR);
	if (res < 0) {
		printf("Error:  Unable to set I2C device to slave address\n");
		exit(EXIT_FAILURE);
	}

	//Configure the I2C-2 bus
	unsigned char buffer[2];
	buffer[0] = MOISTURE_SENSOR_ADDR_CONFIG;
	buffer[1] = 0x20;
	res = write(moistureSensorFD, buffer, 2);
	if (res != 2) {
		printf("Error: Unable to write to I2C register. Expected %d, but received
%d\n", 2, result);
```

```
                exit(EXIT_FAILURE);
        }

        return moistureSensorFD;
}
```

4.2 Reading the Moisture Sensor via the I2C ADC

The following function allows the program to read the Moisture Sensor data stored in the I2C-2 register:

```
#define NUM_REGISTERS 2
#define STARTING_REG 0x00

int readMoistureSensor(int fileDesc) {
        if (fileDesc >= 0) {
                unsigned char registerToRead = 0x00;
                char data[NUM_REGISTERS];
                readI2C(fileDesc, STARTING_REG, data, NUM_REG);
                int res = write(moistureSensor, &registerToRead, sizeof(registerToRead));
                if (res != sizeof(registerToRead)) {
                        printf("Error: Unable to write to register during read. Expected %d, but
received %d\n", sizeof(registerToRead), res);
                        exit(EXIT_FAILURE);
                }

                res = read(fileDesc, data, NUM_REGISTERS];
                if (res != NUM_REGISTERS) {
                        printf("Error: Unable to read from I2C register. Expected %d, but
received %d\n", NUM_REGISTERS, res);
                        exit(EXIT_FAILURE);
                }

                int value = ((data[0] & 0x0F) << 8 | (data[1])) & 0xFFF;
                return value;
        }
}
```

4.3 Main Program

The following code drives the program to read the Moisture Sensor:

```
#include <stdio.h>
#include <stdlib.h>
#include <fcntl.h>
#include <unistd.h>
#include <linux/i2c.h>
#include <linux/i2c-dev.h>
#include <sys/ioctl.h>
```

```
#define MOISTURE_SENSOR_ADDR_CONFIG 0x02
#define MOISTURE_SENSOR_ADDR 0x50
#define BB-I2C2 "BB-I2C2"
#define I2C-2 "/dev/i2c-2"
#define SLOTS "/sys/devices/platform/bone_capemgr/slots"
#define NUM_REGISTERS 2
#define STARTING_REG 0x00

//Put the above functions here

int main() {
	int moistureSensorFD = initializeI2CBus();
	int moistureValue = readMoistureSensor(moistureSensorFD);

	printf("Current moisture value: %d\n", moistureValue);
	close(moistureSensorFD);
	return 0;
}
```