

Laboratory 8: Custom Firmware for xDot
Instructor: Young H. Cho T.A.: Arthur Win
Due Oct 29 at 11:59 PM (Report), Oct 30 at 11:59 PM (Video)

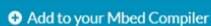
1. Use the USB lead to connect the xDot to the PC. The status light (LED 3) will turn on, indicating it has power.

On Linux systems the following ports are associated with xDot
`/dev/ttyACM(X)`

On Windows systems COM ports appear in the device manager as
Debug Port: Mbed Serial Port
AT Command Port: XR21V1410 USB UART
You may need to install a driver for the debug port to function properly.
<https://developer.Mbed.org/handbook/Windows-serial-configuration>

On Mac, COM ports appear in the device manager as:
`/dev/cu.usbmodem(X)`

2. Go to <http://www.mbed.com> and create an account.
Under Hardware->Boards, search for xdot and select the Multitech xDot board
You might be redirected to a site where you can find a button as shown below,



3. Scroll down to table of contents tab on the right and select 12.

 mbed-os-example-blinky

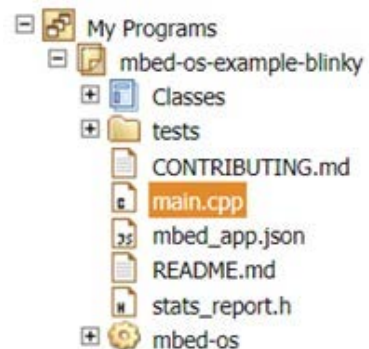
[Import program](#)

This is a very simple guide, reviewing the steps required to get Blinky working on an Mbed OS platform.

Last commit 13 days ago by [mbed-os-examples](#)

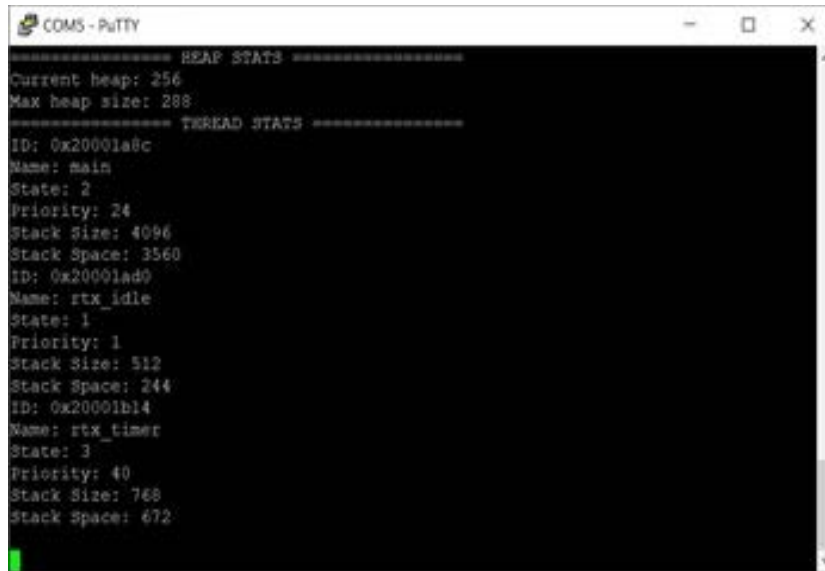
4. Click on import program and a new window with an IDE pops up. Give the project a name and click on Import. The navigator will be populated as so:

5. The files include libraries required to drive and run the xDot. The lib mbed-os provides abstract functions to communicate with xDot. The file main.cpp contains a hello world example code, where it blinks LED1 with a 500ms delay and reports the stats every 10 seconds



6. Compile the code and copy the downloaded bin to the xDot.
7. Once the transmission to xDot is complete (make sure the transmission did not fail) hit the reset button on the xDot.
8. Stats are sent to the debug port
Download a terminal program like Putty, Teraterm, or minicom on Windows or use "cu" on Linux to connect to xDot's debug port through UART
<https://www.cyberciti.biz/hardware/5-linux-unix-commands-for-connecting-to-the-serial-console/>

Specify the baud rate to 115200, Data bits to 8, Parity to N, Stop bits to 1, and Flow Control to OFF



```

COM5 - PuTTY
***** HEAP STATS *****
Current heap: 256
Max heap size: 288
***** THREAD STATS *****
ID: 0x20001a8c
Name: main
State: 2
Priority: 24
Stack Size: 4096
Stack Space: 3560
ID: 0x20001ad0
Name: rtx_idle
State: 1
Priority: 1
Stack Size: 512
Stack Space: 244
ID: 0x20001b14
Name: rtx_timer
State: 3
Priority: 40
Stack Size: 768
Stack Space: 672

```

Figure 1: Monitoring debug port using Putty on Windows

9. Review the following web page for building thermistor
<https://learn.adafruit.com/thermistor/using-a-thermistor>

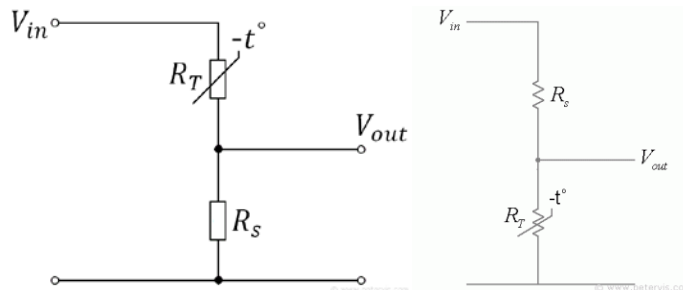


Figure 2: Examples of NTC thermistor circuits

You may purchase parts from Amazon (fast delivery) or other electronic component stores to build a thermistor for xDot. You should consider purchasing other components for your sensor capabilities for the step 12 of this lab as well as your final project.

Keep in mind that the upper voltage limit should be 3.3V for V_{out} .

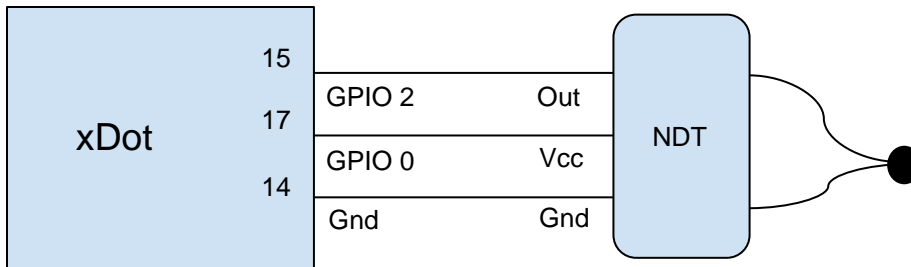
10. Use the example code below to capture data from GPIO2(PB_0) ADC and GPIO 0 as source for the sensor.

The parameter values should be adjusted based on your specific configuration for the thermistor.

Refer to the following document for details of gpio or pin-diagrams of the xDot:

<https://www.multitech.com/documents/publications/manuals/s000645.pdf>

The connections are shown below:



// Example Code

```
#include "mbed.h"
#define BETA 3975
AnalogIn sensor(PB_0);
DigitalOut vcc(GPIO0);
double Thermistor(int RawADC) {
    double Temp;
    Temp= (float) 10000.0 * ((65536.0 / RawADC) - 1.0);
    Temp = (1/((log(Temp/10000.0)/BETA) + (1.0/298.15)));
    Temp = Temp - 273.15; // Convert Kelvin to Celcius
    //Temp = (Temp * 9.0)/ 5.0 + 32.0; // Convert C to F
    return Temp;
}

int main() {
    printf("\r\nTemperature Test program");
    printf("\r\n*****\r\n");
    vcc = 1;
    unsigned int val;
    while (1) {
        val=sensor.read_u16();
        double temp = Thermistor(val);
        printf("%f\n",temp);
        wait(5);
    }
}
```

The function Thermistor() converts the analog value to a readable temperature value scaled on the celsius scale using information from the NTC sensor documentation.

For additional information on mBed OS review the following link:

<https://os.mbed.com/docs/mbed-os/v5.14/introduction/index.html>

11. Compile the code and download it to the xDot.

Open a serial connection to debug port with a baud of 115200 and check for the sensor value correctness or calibrate accordingly.

```
Temperature Test program
*****
23.466306
    23.444667
        23.444667
            23.466306
```

12. Dig in and experiment with one additional sensor of your choice, sampling rates, and GPIO capabilities. One easy and interest experiment might be to get a 3.5mm audio/mic connector with wires to feed in your phone audio output to xDot analog input and see if you can capture or detect sound.

<https://www.amazon.com/Qaoquda-Balanced-Terminal-Adapter-Connector/dp/B07RXS3ZYN>