

How to Set Up MNCVSM

ECE4012 Senior Design Project

Section L2A, Wildcats
Project Advisor, Dr. Ying Zhang

Nydrel Jack, njack7@gatech.edu
Arianne Perez, aperez71@gatech.edu
Chelsi Taylor, ctaylor63@gatech.edu
Ethan Vargas, evargas30@gatech.edu
Nathanael Williams, nwilliams68@gatech.edu

Table of Contents

Repositories	2
Installing the MNCVSM Mobile Application	2
Configuring the Bluetooth Module	3
Flashing the Project onto the MCU	4
Wiring the MCU	5
Establishing a Bluetooth Connection	6
Running the Mobile Application	7
Looking at the DSP Algorithm	7

Repositories

Below are the three repositories that contain the code used to construct the Mobile Non-Contact Vital Sign Monitoring (MNCVSM) project.

Repository containing the Android Studio project for the Android application:

<https://github.gatech.edu/njack7/app-mncvsm>

Repository containing the μ Vision project for the STM32373C MCU:

<https://github.gatech.edu/njack7/keil-mncvsm>

Repository containing the code for the digital signal processing algorithm:

<https://github.gatech.edu/njack7/mcu-mncvsm>

Installing the MNCVSM Mobile Application

There are two methods that can be used to get the MNCVSM mobile application onto an Android device.

Note: The Android device must have an API level of 21 or higher (Android versions 5.0 or greater).

Method 1

On your mobile device, download and install the application's APK from here:

https://drive.google.com/open?id=1Gb2OnD2sd35Uhn0sURBQIYh6w58_VaVf

Note: You may be prompted to temporarily allow your mobile device to install applications from insecure sources.

Method 2

Download and install the latest version of Android Studio to your computer. Next, import the mobile application project from the repository linked above by going to File > New > Project from Version Control > Git and typing "https://github.gatech.edu/njack7/app-mncvsm.git" for the Git Repository URL.

Once the project has been imported and built (a notification will appear letting you know when the project has built successfully), connect your mobile device to your computer and run the application on your device by pressing the green play button and selecting your connected device.

Once the application is running on your mobile device, you may disconnect it from your computer and the application will appear amongst your other apps.

Configuring the Bluetooth Module

The Bluetooth module being used in this project is the RN-41 Class 1 Bluetooth Module. The one used in the MNCVSM project has already been configured to have a customized name and password (“mncvsm” and “wildcats12a”). If the module must be reconfigured or a new module must be configured for the first time, you must establish a serial connection with the module through its UART interface. Through this connection, you must be able to send a series of commands to change the module’s configuration settings. Our team did this by using an mbed (<https://www.mbed.com/en/>) and connecting it to the Bluetooth module using the following connections:

mbed	RN-41 Bluetooth Module
gnd	gnd
3.3V	Vin (3.3V)
not connected	RTS
gnd	CTS
P10	TX
P9	RX

The mbed was then flashed with the following program:

```
#include "mbed.h"
RawSerial pc(USBTX, USBRX);
RawSerial dev(p9,p10);
DigitalOut led1(LED1);
DigitalOut led4(LED4);
void dev_rcv()
{
    led1 = !led1;
    while(dev.readable()) {
        pc.putc(dev.getc());
    }
}
void pc_rcv()
{
    led4 = !led4;
    while(pc.readable()) {
        dev.putc(pc.getc());
    }
}
int main()
{
    pc.baud(115200);
    dev.baud(115200);
    pc.attach(&pc_rcv, Serial::RxIrq);
    dev.attach(&dev_rcv, Serial::RxIrq);
    while(1) {
        sleep();
    }
}
```

While this program was running (and the mbed was connected to a computer), TeraTerm was opened and connected to the mbed's COM port. The baudrate of this connection was also set to 115200. This allowed us to send commands through the mbed to the Bluetooth module.

While this connection was active, sending the command “\$\$\$” will enter Command Mode, where you can change configuration settings. The following two commands were used to set the username and password, respectively:

```
s-,mncvsm
```

```
sp,wildcatsl2a
```

For more information on the Bluetooth module and ways to configure it, see:

https://cdn.sparkfun.com/datasheets/Wireless/Bluetooth/bluetooth_cr_UG-v1.0r.pdf.

Note: A configured Bluetooth module is not necessary to run the completed project. See the “Establishing a Bluetooth Connection” section for details.

Flashing the Project onto the MCU

Download the μ Vision project from the repository linked above (Click “Clone or download” > Download ZIP and unzip the downloaded file) and open the project file (found in the MDK-ARM directory). With the MCU connected to your computer, try to “Rebuild” the project and then “Download” it onto the device. If the LEDs on the device blink momentarily, that means that the program has been flashed onto the MCU. Press the “Reset” button on the device to begin running the program.

If you do not have the appropriate software pack for the device, you may be prompted to download it. If not immediately redirected, navigate to the link below and download the STMicroelectronics STM32F3 Series Device Support under “Software Pack.”

https://www.keil.com/boards2/stmicroelectronics/stm32373c_eval/

Wiring the MCU

In order for the system to work, the MCU must be connected to both the Bluetooth module and the set-up sensor system. Below are the pin connections that are needed between the MCU's evaluation board and the Bluetooth module.

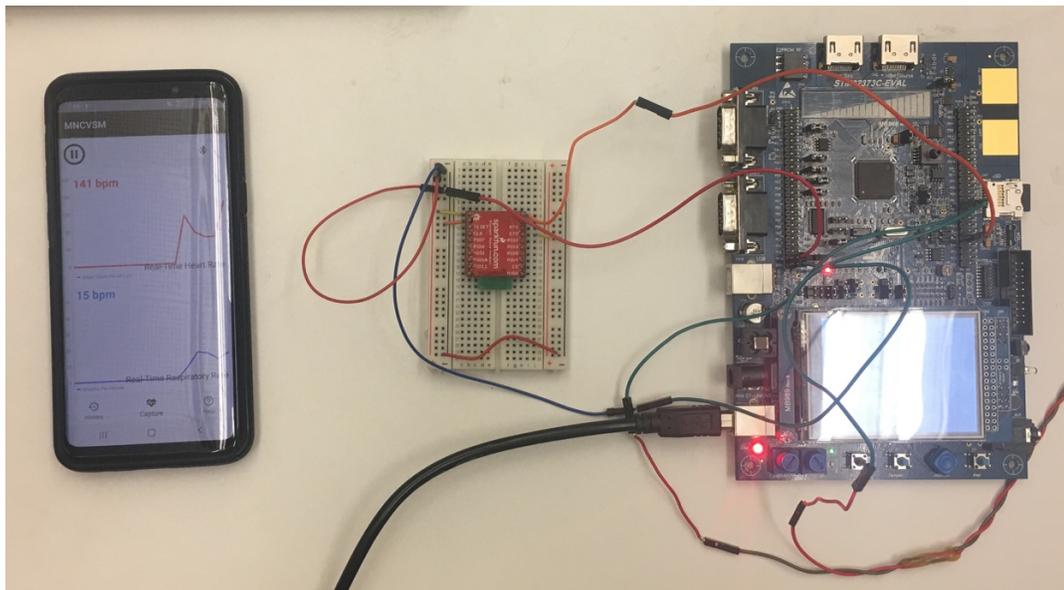
STM32373C Evaluation Board	RN-41 Bluetooth Module
gnd	gnd
3V3	Vin (3.3V)
not connected	RTS
gnd	CTS
not connected	TX
Pin A2	RX

To connect the MCU to the sensor system, a high frequency SMA to PCB Mount connector was used. The SMA connector was connected to one of the sensor system's outputting SMA cables while two wires were soldered to the center pin and surrounding ground pins of the PCB Mount. This soldering is shown below.



The wire connected to the center pin is then connected to Pin B0 on the evaluation board while the wire connected to the ground pin is connected to Pin B1.

The system is now fully connected, as shown below.



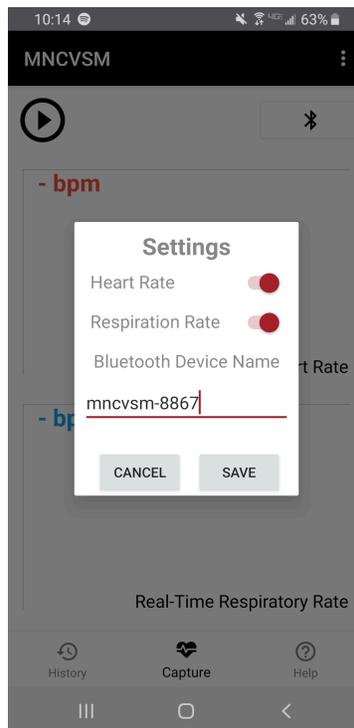
Establishing a Bluetooth Connection

In order to establish a Bluetooth connection between the Bluetooth module and the mobile device, first ensure that the MCU is powered on (this will mean that the Bluetooth module is powered on, if the previous connections between the MCU and the module are made). Next, pair your mobile device with the Bluetooth module by navigating to your phone's Bluetooth settings and finding the device underneath "Available devices."

Note: The Bluetooth module's name will be the name that you set during the configuration stage followed by a series of numbers. This series of number is part of the module's address and is automatically added at the end of the user-designated name. The name of the device set up by our team is "mncvsm-8867."

After clicking on the device's name and providing the PIN if prompted ("wildcats12a" if configured, "1234" by default), your mobile device should be paired with the Bluetooth module.

Next, open the downloaded MNCVSM application. You may set up a password for the application or simply press "skip" to ignore authentication features. Open the application's settings by clicking the "..." at the top right of the screen. Double check that the "Bluetooth Device Name" matches the one that you have paired with. **If you are using a new Bluetooth module or have configured the module to have a different name, change this device name and press "Save."**



Finally, press the Bluetooth symbol near the top right of the application and wait for the application to connect. If an error is returned, try reconnecting a few more times.

Running the Mobile Application

After making sure that the MCU is powered on and connected properly, your application has connected with the Bluetooth module, and the sensor system is operating, you can now begin using the mobile application to observe a user's vital sign signals. After having a user sit in front of the sensors and stay still, press the "Play" button in the application to begin the process. Every 35 seconds, an updated value for the calculated heart and respiratory rates will be shown on the screen.

Note: The very first value received may be inaccurate due to the user getting settled in front of the sensors.

When you are finished collecting vital sign readings, you may stop the process by clicking the "Pause" button. After which, you can continue capturing vital sign readings by pressing "Play" again, save the latest value by pressing "Download," or clear all shown values by pressing "Close."

Additional features within the application include a "History" page that shows saved respiratory and heart rates and a "Help" page that includes a basic tutorial on how a user could use the system.

Looking at the DSP Algorithm

To take a closer look at the digital signal processing algorithm that is being used to determine the respiratory and heart rates, open up the digital signal processing repository linked above. The latest versions of the code are in the "C code 04_20_2019" and "C++ code 04_20_2019" directories, containing the C and C++ versions of the algorithms, respectively. However, the final code used a downsampling ratio of 32, instead of the ratio of 16 seen in the repository. This final code can be found in the "function.c" and "function.h" files that exist within the MCU μ Vision project.