



NPTEL ONLINE CERTIFICATION COURSES

Course Name: Embedded System Design with ARM

Faculty Name: Dr. Kamalika Datta

Department : Computer Science and Engineering

Topic

Lecture 38: Accelerometer

CONCEPTS COVERED

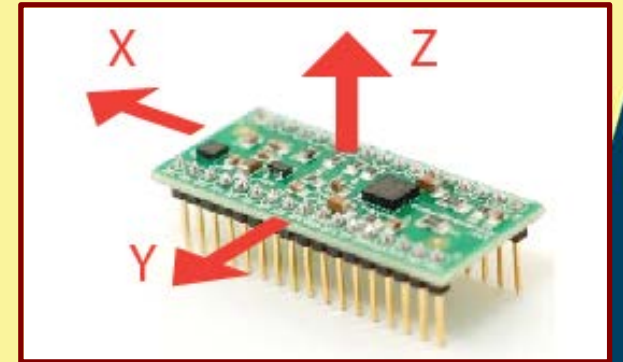
- ❑ Accelerometer: principle of operation
- ❑ ADXL 335 accelerometer



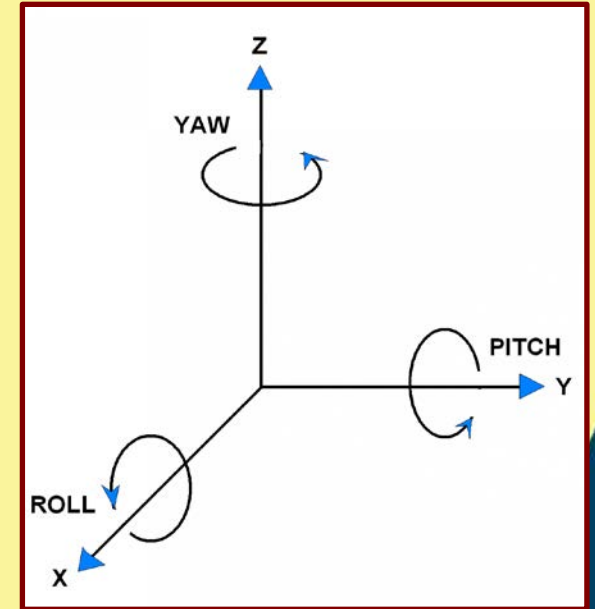
Sensor: Accelerometer (Motion)

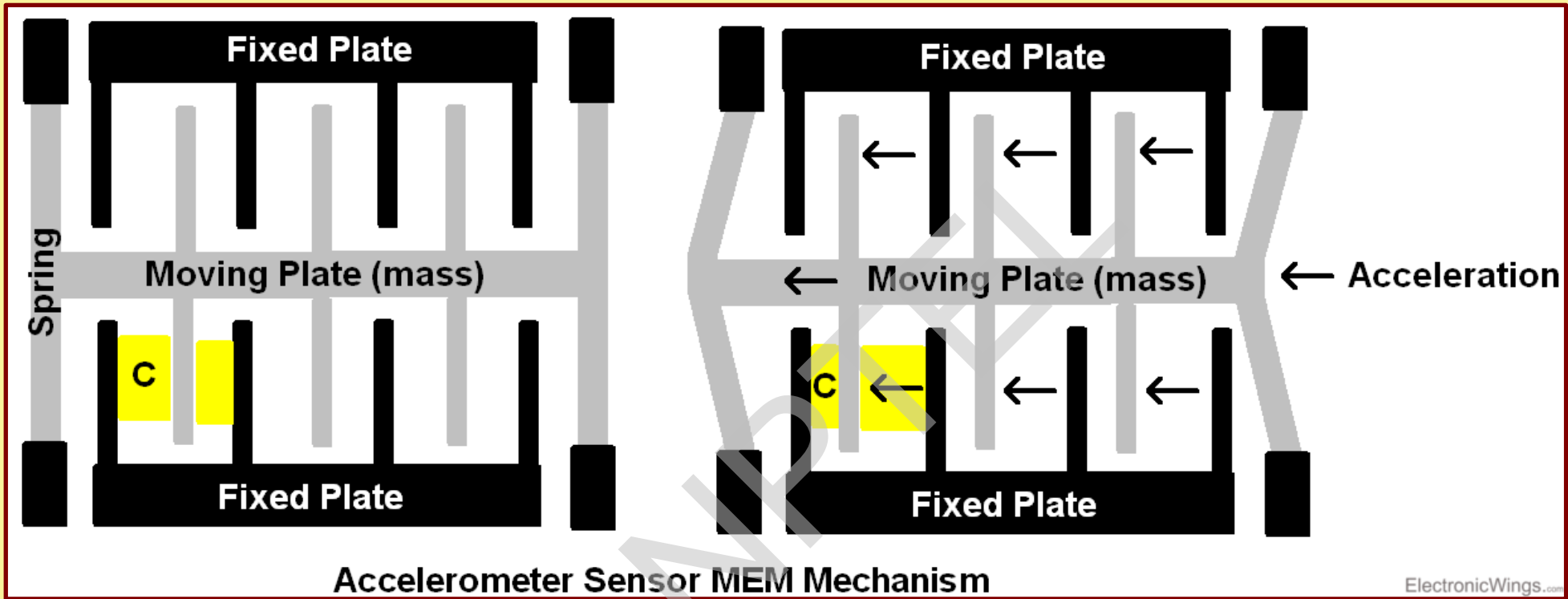
What is an accelerometer?

- A dynamic sensor that can measure acceleration in one, two, or three orthogonal axes.
- Typically used in one of three modes:
 - As an inertial measurement of velocity and position.
 - As a sensor of inclination, tilt or orientation in 2/3 dimensions.
 - As a vibration or impact (shock) sensor.
- Most accelerometers are based on Micro Electro-Mechanical Sensors (MEMS).
 - Based on the displacement of a small mass etched into the silicon surface of the IC, and suspended by small beams.
 - As an acceleration is applied, a force develops ($P = mf$) that displaces the mass.



- The displacement of the mass can be measured using capacitive sensing or piezoelectric effect sensing.
 - Change in capacitance, or generation of a voltage.
- The basic structure of an accelerometer consists of fixed plates and moving plates (called *mass*).
 - Acceleration deflects the moving mass and unbalances the differential capacitor that results in a sensor output voltage amplitude which is proportional to the acceleration.
 - By measuring the acceleration along the x, y and z directions, it is possible to calculate the inclination or tilt.
 - It is also possible to calculate the angles of rotation along x, y and z axes (called *roll*, *pitch* and *yaw* respectively).



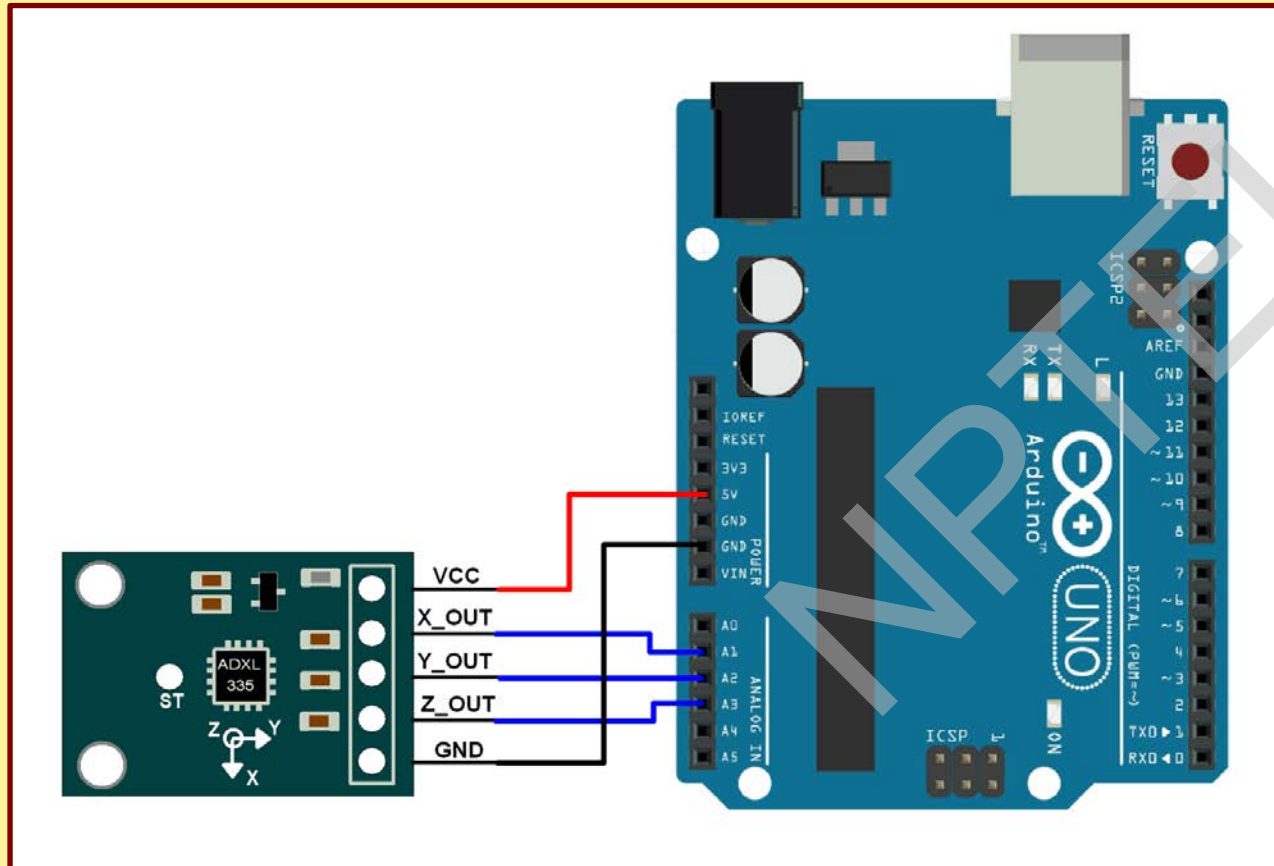


The ADXL 335 Accelerometer

- The ADXL 335 is a small, thin, low-power 3-axis accelerometer with signal conditioned voltage outputs.
- It can measure the static acceleration due to gravity in tilt-sensing applications, as well as dynamic acceleration resulting from motion, shock, or vibration.
- The module measures acceleration within range $\pm 3g$ in the x, y and z axes.
 - The output signals (X_OUT, Y_OUT, Z_OUT) are analog voltages that are proportional to the acceleration.



Interfacing ADXL 335 Accelerometer to Arduino UNO



X_OUT, Y_OUT and Z_OUT are connected to analog input pins.



NPTEL ONLINE CERTIFICATION COURSES

**Thank
you!**



NPTEL ONLINE CERTIFICATION COURSES

Course Name: Embedded System Design with ARM

Faculty Name: Dr. Kamalika Datta

Department : Computer Science and Engineering

Topic

Lecture 39: Experiment using Accelerometer

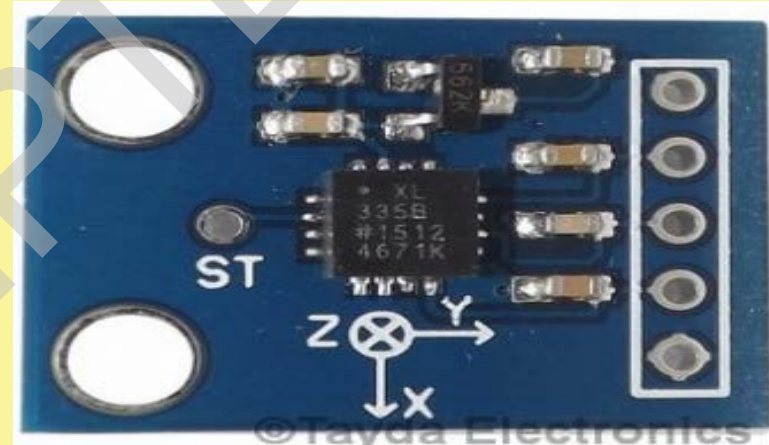
CONCEPTS COVERED

- ☐ Experiment with Accelerometer
- ☐ Demonstration

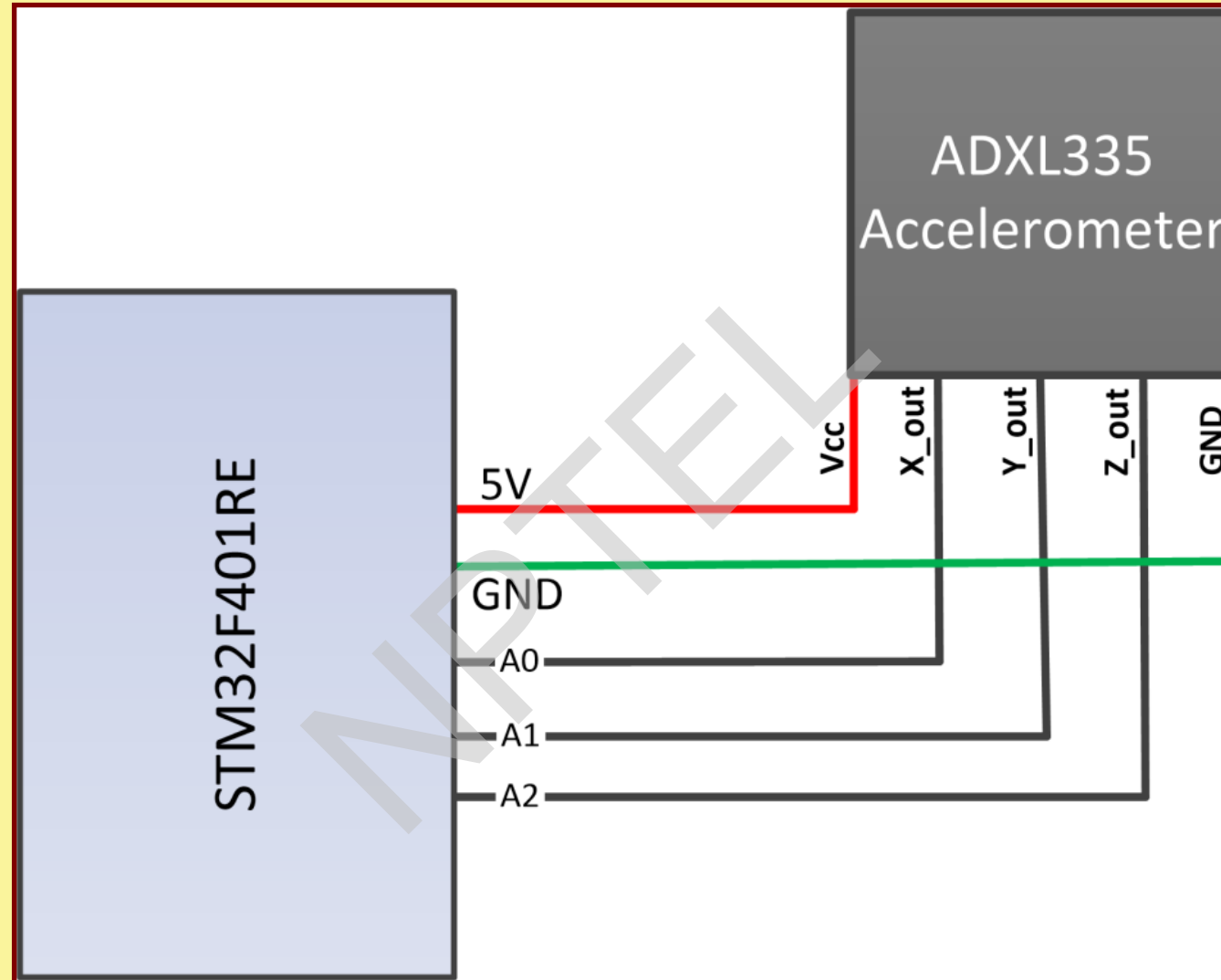


ADXL335 and STM kit connection

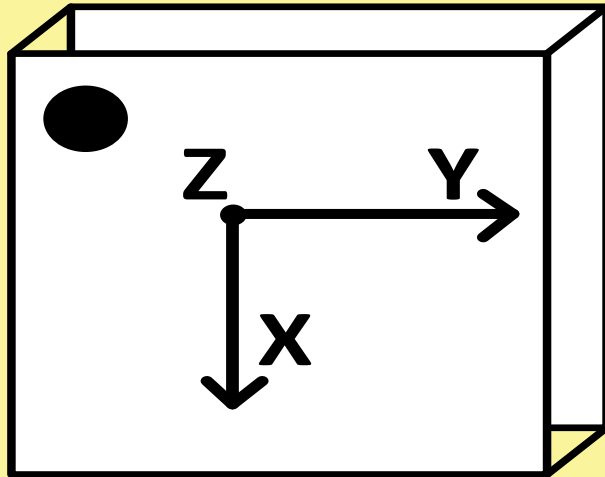
- Vcc – voltage source (+5V)
- GND – to ground
- X_out, Y_out, Z_out – Analogue input pins A0, A1, A2.



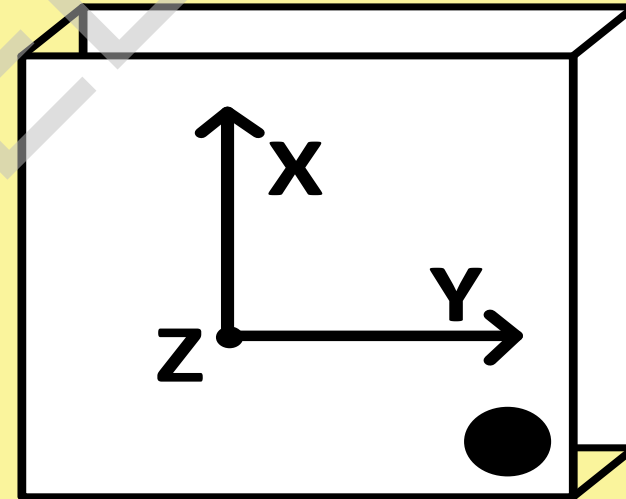
Connection Diagram



Analysis for movement along x-axis

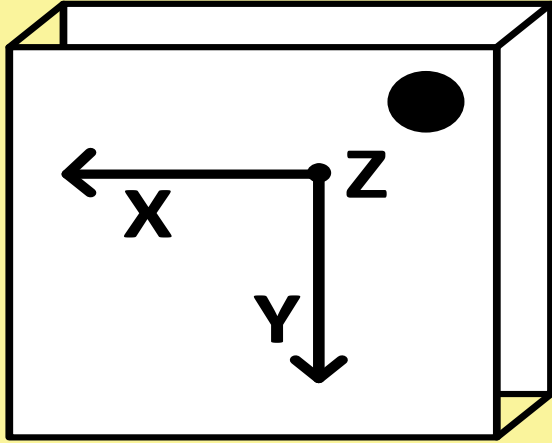


X value min

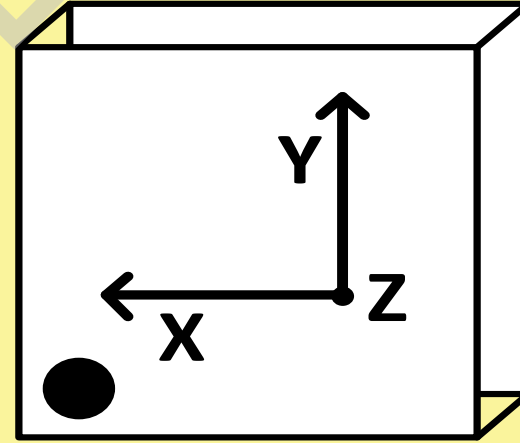


X value max

Analysis for movement along y-axis

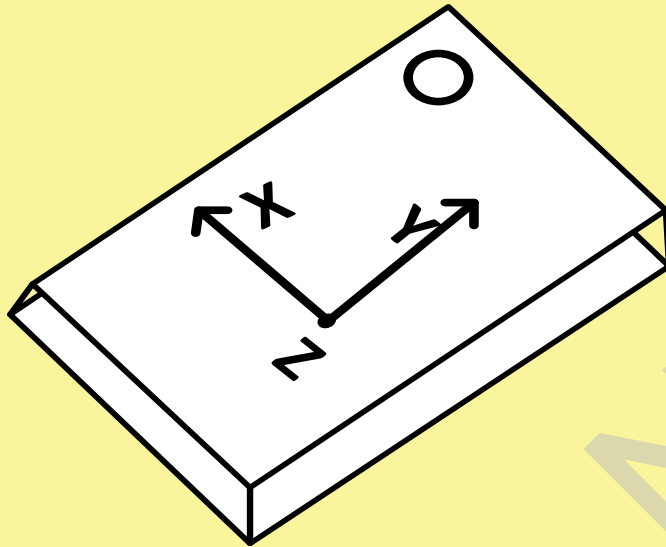


Y value min

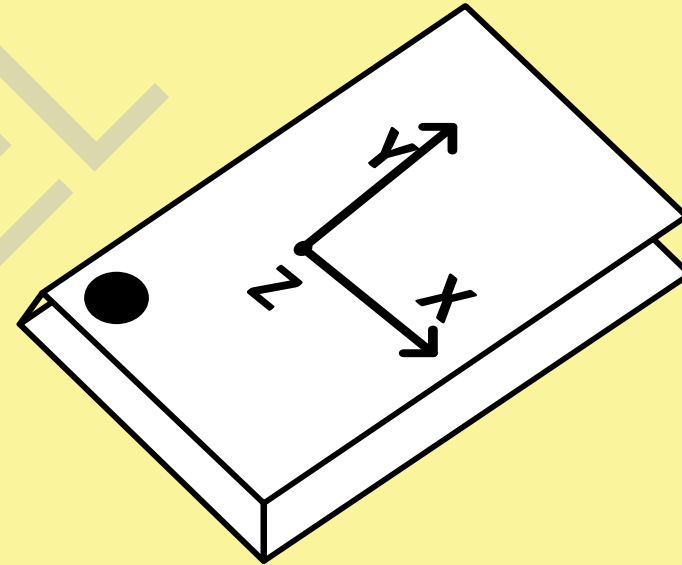


Y value max

Analysis for movement along z-axis



Z value min



Z value max

Mbed C Code for Analysis

```
#include "mbed.h"
Serial pc (USARTX, USARTY);
AnalogIn x (A0);
AnalogIn y (A1);
AnalogIn z (A2);
int main() {
    int x1, y1, z1;
    while(1) {
        x1 = x.read_u16();
        y1 = y.read_u16();
        z1 = z.read_u16();
        pc.printf ("%d,%d,%d\n", x1, y1, z1);
        wait(1);
    }
}
```

Printing Data to MATLAB

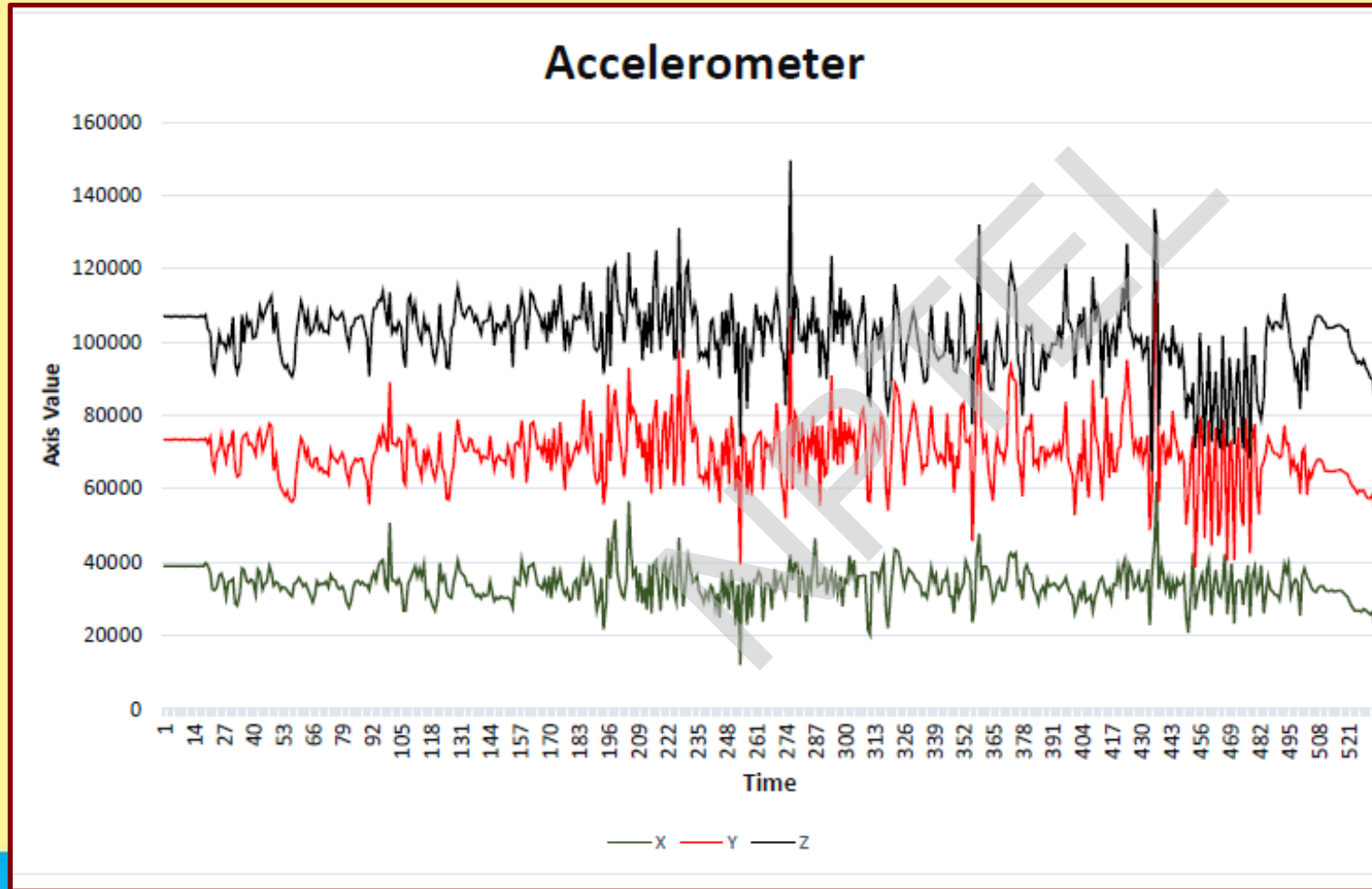
- The printing of data into MATLAB is similar to that for HyperTerminal software.
- MATLAB also uses serial connection to get data from serial port.
- The mbed code for reading the data from the STM32 kit is the same as shown before.

Complete MATLAB Code

```
function plotting()  
TIMEOUT = 5; % time to wait for data before aborting the connection  
XPOINTS = 100; % no of points along x axis  
try  
    mbed = serial('COM3','BaudRate',9600,'Parity','none','DataBits',  
                8,'StopBits',1);  
    set(mbed,'Timeout',TIMEOUT);  
    fopen(mbed);  
    position = 1;  
    time = 1;  
    x = [(1:XPOINTS)' (1:XPOINTS)' (1:XPOINTS)'];  
    xlabel = (1:XPOINTS);  
    y = zeros(XPOINTS,3);  
    flushinput(mbed);
```

```
while (1)
    values = fscanf(mbed, '%d,%d,%d\n');
    y(position,:)=values';
    xlables(position)=time;
    time=time+1;
    if (position < XPOINTS)
        position = position + 1;
    else
        position = 1;
    end
    plot(x,y);
    drawnow;
end
fclose(mbed);
catch
    disp('Failed!');
    fclose(mbed);
end
```

Sample Output Waveform





NPTL ONLINE CERTIFICATION COURSES

Thank
you!



NPTEL ONLINE CERTIFICATION COURSES

Course Name: Embedded System Design with ARM

Faculty Name: Dr. Kamalika Datta

Department : Computer Science and Engineering

Topic

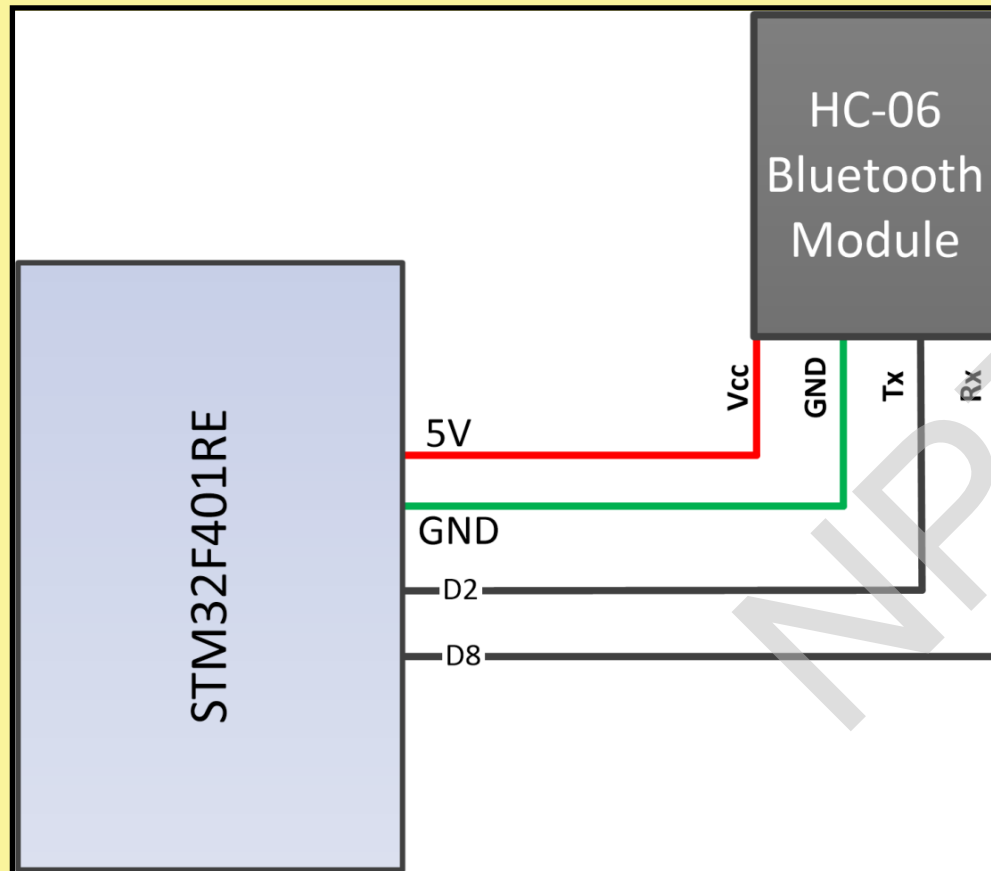
Lecture 40: Experiment using Bluetooth

CONCEPTS COVERED

- ❑ Experiment with Bluetooth and accelerometer
- ❑ Demonstration



Connection Diagram



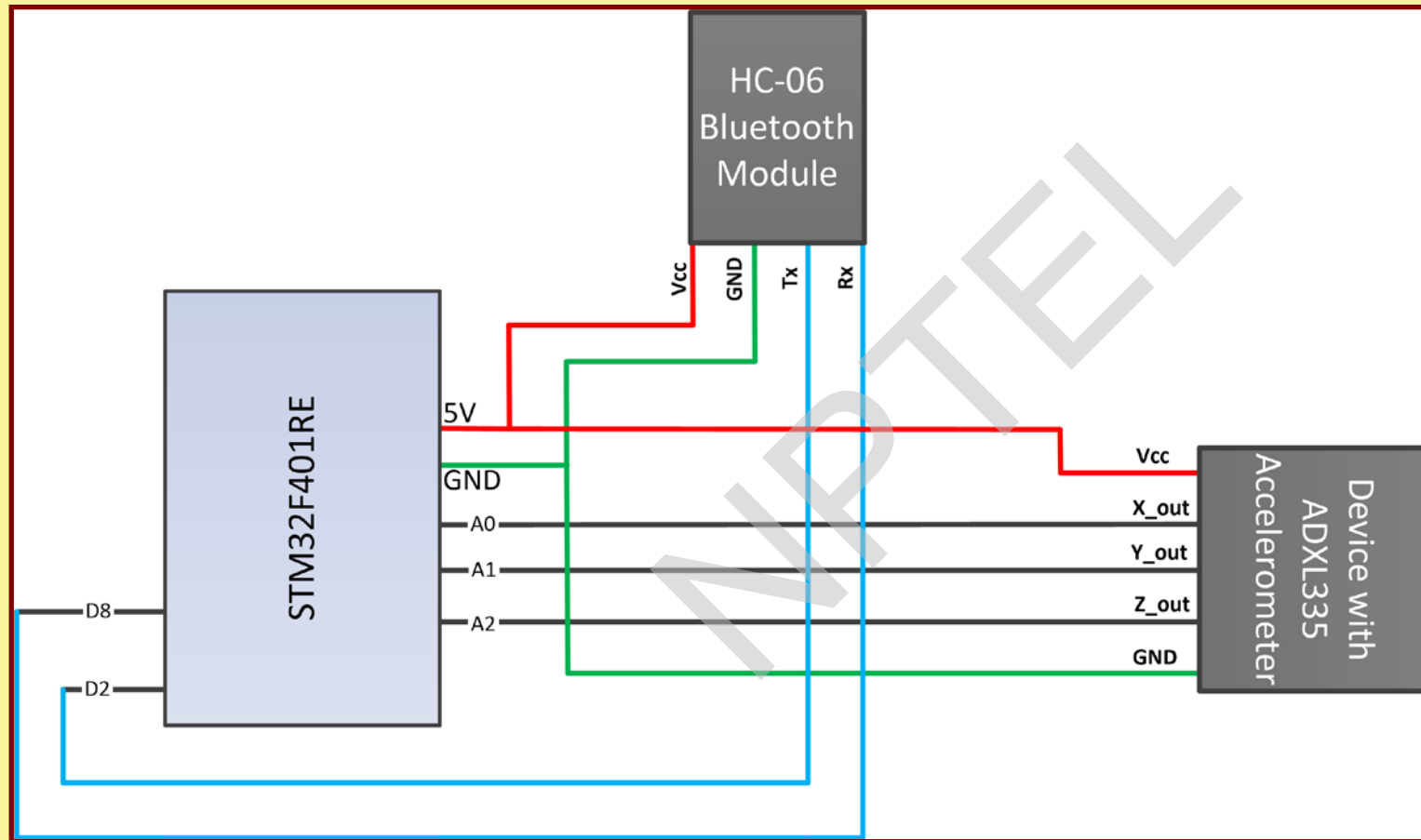
- After this connection LED of Bluetooth module HC-06 will start blinking.
- When the device is connected with any other Bluetooth device the it will stop blinking.

Test Program to Print using Bluetooth Module

```
#include "mbed.h"
Serial bluetooth(D8, D2); //RX, TX

int i=0;
int main()
{
    bluetooth.printf("Hello\n\n");
    wait(5);
    bluetooth.printf("Welcome to\n");
    bluetooth.printf("EMBEDDED SYSTEM DESIGN WITH ARM");
    bluetooth.printf("NPTEL-2018");
}
```

Interfacing Bluetooth and Accelerometer Modules



Mbed C Program – Version 1

```
#include "mbed.h"
#include <string.h>
Serial bluetooth(D8, D2); //RX, TX
AnalogIn X(A1);
AnalogIn Y(A2);
AnalogIn Z(A3);
int main()
{
    int x1, y1, z1;
    while(1) {
        x1 = X.read_u16();
        y1 = Y.read_u16();
        z1 = Z.read_u16();
```

```
if (x1>28000 && x1<31000 && y1>35000 && y1<39000 && z1>35000 && z1<39000)
{
    bluetooth.printf ("Device is vertically up \n\n");
    wait(1);
}
else if (x1>42000 && x1<46000 && y1>34000 && y1<39000 && z1>34000 &&
        z1<39000)
{
    bluetooth.printf ("Device is vertically down \n\n");
    wait(1);
}
else if (x1>35000 && x1<39000 && y1>35000 && y1<39000 && z1>43000 &&
        z1<46000)
{
    bluetooth.printf ("Device is flat \n\n");
    wait(1);
}
```

```

else if (x1>34000 && x1<39000 && y1>26000 && y1<30000 && z1>34000 &&
        z1<39000)
{
    bluetooth.printf ("Device is horizontally left \n\n");
    wait(1);
}
else if (x1>34000 && x1<39000 && y1>43000 && y1<46000 && z1>34000 &&
        z1<39000)
{
    bluetooth.printf ("Device is horizontally right \n\n");
    wait(1);
}
wait(2);
}
}

```

Mbed C Program – Version 2

```
#include "mbed.h"
#include <string.h>
Serial bluetooth(D8, D2); //RX, TX
AnalogIn X(A1);
AnalogIn Y(A2);
AnalogIn Z(A3);
int main(){
    int flag=0;
    int old_flag=0;
    int x1, y1, z1;
    char buf[50];
    while(1) {
        x1 = X.read_u16();
        y1 = Y.read_u16();
        z1 = Z.read_u16();
```

```

if (x1>28000 && x1<31000 && y1>35000 && y1<39000 && z1>35000 && z1<39000) {
    flag=1;
    sprintf (buf, "%s", "Device is vertically up \n\n");
}
else if (x1>42000 && x1<46000 && y1>34000 && y1<39000 && z1>34000 &&
        z1<39000) {
    flag=2;
    sprintf(buf, "%s", "Device is vertically down \n\n");
}
else if (x1>35000 && x1<39000 && y1>35000 && y1<39000 && z1>43000 &&
        z1<46000) {
    flag=3;
    sprintf(buf, "%s", "Device is flat \n\n");
}
else if (x1>34000 && x1<39000 && y1>26000 && y1<30000 && z1>34000 &&
        z1<39000) {
    flag=4;
    sprintf(buf, "%s", "Device is horizontally left \n\n");
}

```

```

else if (x1>34000 && x1<39000 && y1>43000 && y1<46000 && z1>34000 &&
        z1<39000) {
    flag=5;
    sprintf(buf, "%s", "Device is horizontally right \n\n");
}
if (flag != old_flag) {
    old_flag = flag;
    bluetooth.printf ("%s", buf);
    wait(1);
}
wait(1);
}
}

```



NPTEL ONLINE CERTIFICATION COURSES

Thank
you!



NPTEL ONLINE CERTIFICATION COURSES

Course Name: Embedded System Design with ARM

Faculty Name: Dr. Kamalika Datta

Department : Computer Science and Engineering

Topic

Lecture 41: Experiment with Gas Sensor

CONCEPTS COVERED

- ☐ Interfacing the MQ2 gas sensor to STM32 board
- ☐ Demonstration



About the MQ2 Gas Sensor

- The MQ-2 gas sensor is most suited to measure the presence of the following gases:
 - Methane, Butane, LPG, Smoke
- Pin configuration:
 - Pin 1: Power supply voltage (typically +5V)
 - Pin 2: Ground connection
 - Pin 3: Digital Out – We can get a digital output from this pin, by setting a threshold value using the potentiometer.
 - Pin 4: Analog Out – This pin outputs 0-5V analog voltage based on the intensity of the gas.



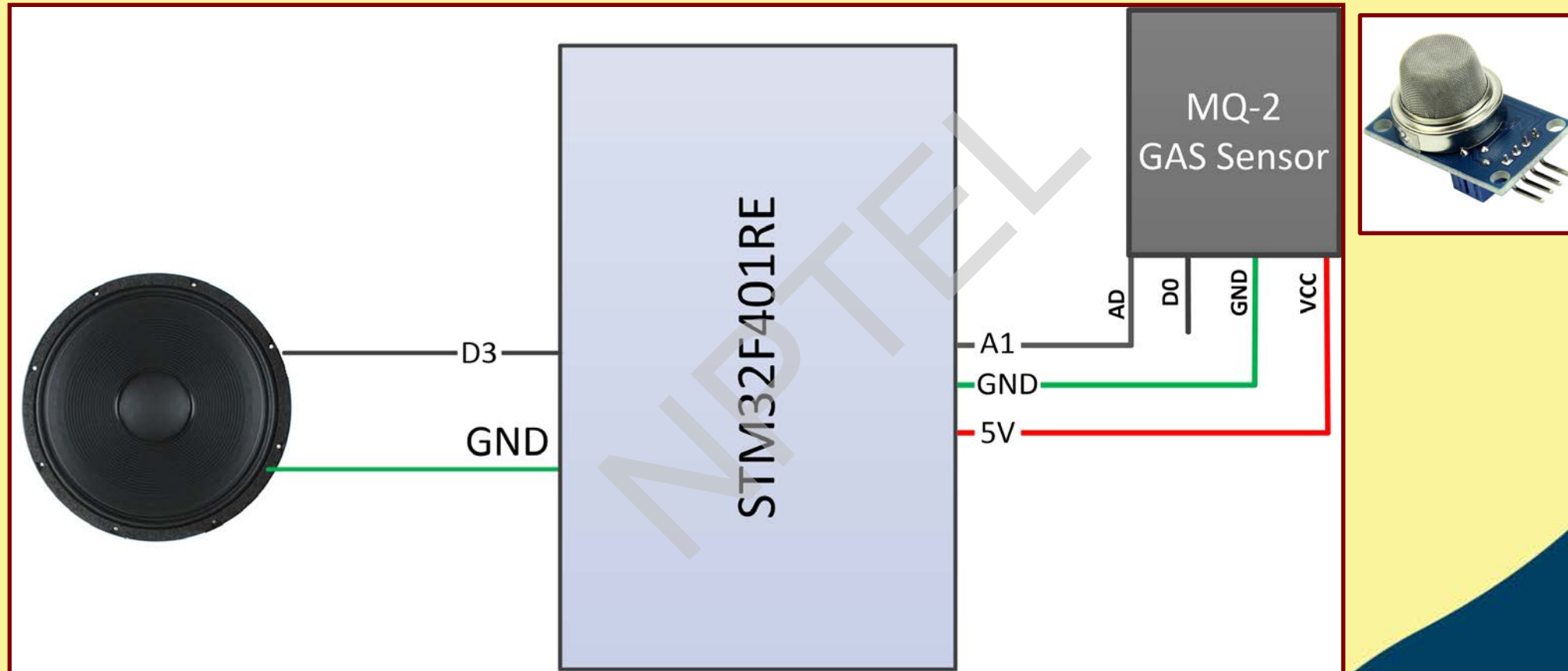
- Point to note:

- There is a heater inside the sensor, which requires 20 seconds to preheat after power is applied.
- For interfacing with the microcontroller, the Analog Out pin can be connected to one of the analog input pins.
 - A suitable threshold can be set to trigger an alarm on detecting some gas.

The Experiment

- When the presence of gas is detected, an audible alarm has to be sounded.
 - The MQ-2 gas sensor is connected to the analog input pin A1 of the STM32F501 Nucleo board.
 - A speaker is connected to the PWM digital output pin D3.

Connection Diagram – STM32



Mbed C Code

```
#include "mbed.h"
PwmOut mypwm(D3);
AnalogIn G(A1);
int i=0;

void fire(){
    i=0;
    while (i<5) {
        mypwm.period_us (3000);
        mypwm.pulsewidth_us (1550);
        mypwm.period_us (3000);
        mypwm.pulsewidth_us (1550);
        wait (0.5);
        mypwm.period_us (2200);
        mypwm.pulsewidth_us (100);
        wait (0.5);
        i++;
    }
}
```

```
int main()
{
    float x;
    while(1) {
        x = G.read();
        printf ("%4.2f\n",x);
        if (x > 0.30)
        {
            fire();
            wait(1);
            mypwm.period_us(10);
            mypwm.pulsewidth_us(5);
        }
    }
}
```



NPTEL ONLINE CERTIFICATION COURSES

**Thank
you!**