**WIFI MODULE:**

In SJ Gateway board, we can detect the COM port for STLink port and we are in the process of updating Wi-Fi firmware to check whether it is getting loaded by the COM port.

**WHEEL CHAIR PROJECT:**

The wheelchair project requires control of DC and Servo motor attached to ForceMech Voyager R2 with the help of keyboard (W, A, S, D). This keyboard input is processed by the STM Discovery Kit B-L475E-IOT01A and PWM signals from the kit is sent to DC Motor.

**Progress: Wheelchair is successfully controlled by ST Discovery board by wire. We have started working to develop code for wireless remote control.**

*Tasks done:*

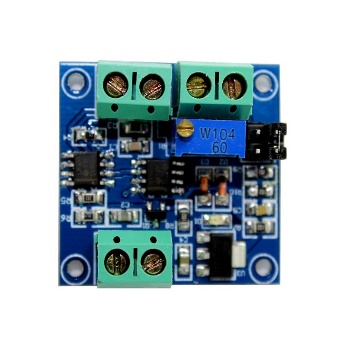
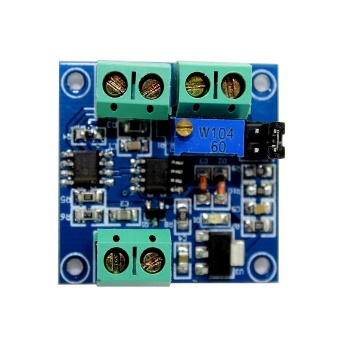
1. The specification of the motor driver used to drive the wheelchair are provided on the following website.

<https://www.bifelectronic.com/gb/others-modules-/2713-0v-to-5-v-0v-to-10-v-0-100-pwm-signal-voltage-convert-pwm-module-bif.html>

1. The connection for the wheelchair remote control and motor driver are as shown below:

PWM-Y VCC- O/p GND PWM-X

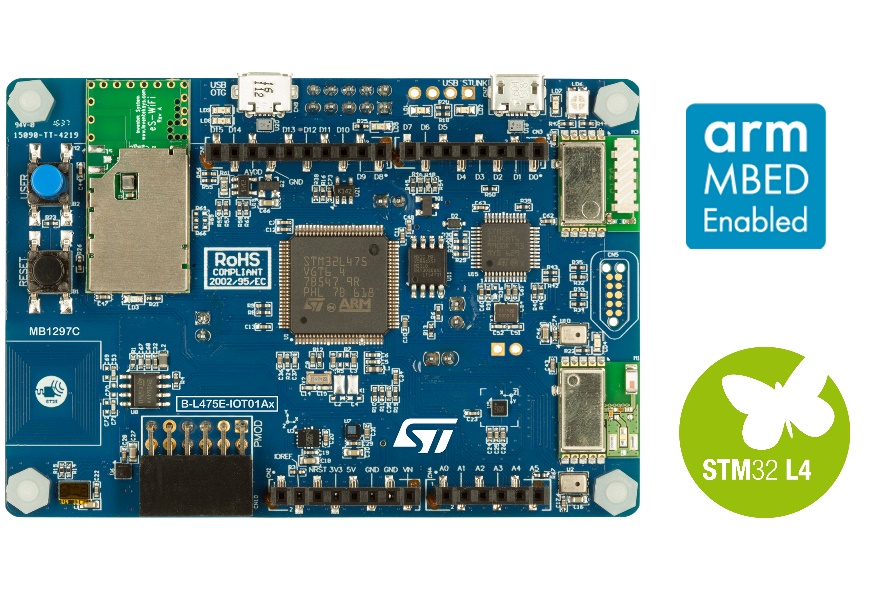




PWM GND GND VIN PWM GND GND VIN

Provide 7.4V at 200mA

GND VCC GND VCC



Keep it open. VCC 5V output from remote

1. Studied the PWM API’s and added the code to set the PWM control on ST Microcontroller.

**void** **pwm\_setvalue**(uint16\_t value\_channel\_1, uint16\_t value\_channel\_4) {

TIM\_OC\_InitTypeDef sConfigOC;

TIM\_OC\_InitTypeDef sConfigOC2;

sConfigOC.OCMode = TIM\_OCMODE\_PWM1;

sConfigOC2.OCMode = TIM\_OCMODE\_PWM1;

sConfigOC.Pulse = value\_channel\_1;

sConfigOC2.Pulse = value\_channel\_4;

sConfigOC.OCPolarity = TIM\_OCPOLARITY\_HIGH;

sConfigOC2.OCPolarity = TIM\_OCPOLARITY\_HIGH;

sConfigOC.OCFastMode = TIM\_OCFAST\_DISABLE;

sConfigOC2.OCFastMode = TIM\_OCFAST\_DISABLE;

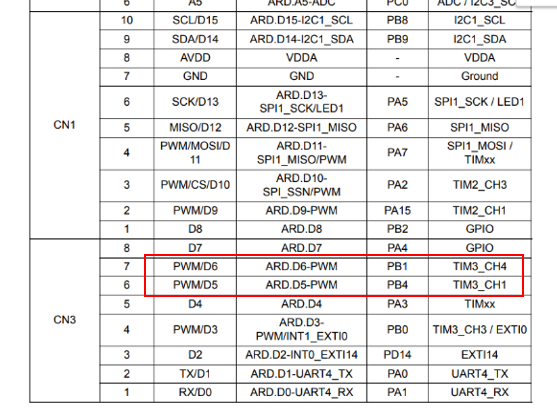
HAL\_TIM\_PWM\_ConfigChannel(&htim3, &sConfigOC, TIM\_CHANNEL\_1);

HAL\_TIM\_PWM\_ConfigChannel(&htim3, &sConfigOC2, TIM\_CHANNEL\_4);

HAL\_TIM\_PWM\_Start(&htim3, TIM\_CHANNEL\_1);

HAL\_TIM\_PWM\_Start(&htim3, TIM\_CHANNEL\_4);

}

1. The pinout for the PWM in ST Discovery kit was found from the schematics of IoT Discovery kit. We made use of PWM3 and TIMER3 with channels 1 and 4 to provide signal to two different motors on the wheelchair. Below is the screenshot. We used pin D5 and pin D6 of the board. Make the ground for the motor driver and the PWM signal.
2. Successfully checked the PWM output using the LED’s connected to STM32 IoT Discovery Kit and test the PWM.
3. From the below Arduino code, Initial values are set for pin 9 and pin 11 at 660 and 665 as initial DC motor and Servo Motor position. The front and back threshold for DC is set at 550 and 760 and left and right movement of servo is set at 550 and 710. By pressing W, A, S, D, the values increase and decreases by 10 and 30 respectively. The Arduino code for the same is as follows:

#include "Keyboard.h"

int Pin1 = 9;

int Pin2 = 11;

int Pin3 = 4;

int Pin4 = 5;

//int analogPin = 3; // potentiometer connected to analog pin 3

int val1 = 660; //680 // variable to store the read value

int val2 = 665;

void setup()

{

pinMode(Pin1, OUTPUT); // sets the pin as output

pinMode(Pin2, OUTPUT);

Serial.begin(9600);

}

void loop()

{

if(Serial.available())

{

char inChar = Serial.read();

switch(inChar)

{

case('w'): val1=val1+30;

if(val1>760)

{

val1=760;

}

break;

case('s'): val1=val1-30;

if(val1<550)

{

val1=550;

}

break;

case('a'):val2=val2-10;

if(val2<550)

{

val2=550;

}

break;

case('d'):val2=val2+10;

if(val2>710)

{

val2=710; }

break;

}

}

Serial.print(val1);

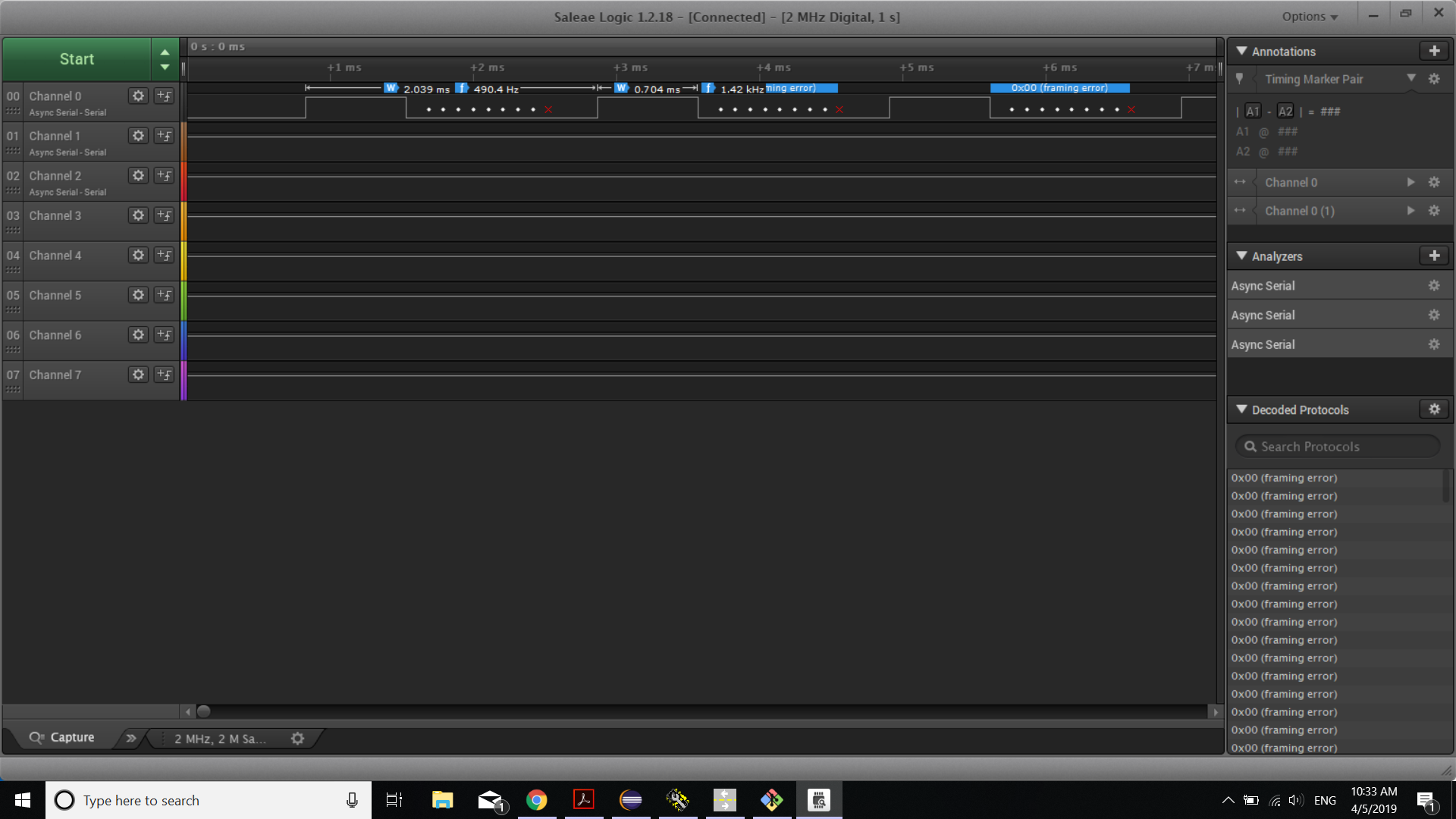
Serial.print(":");

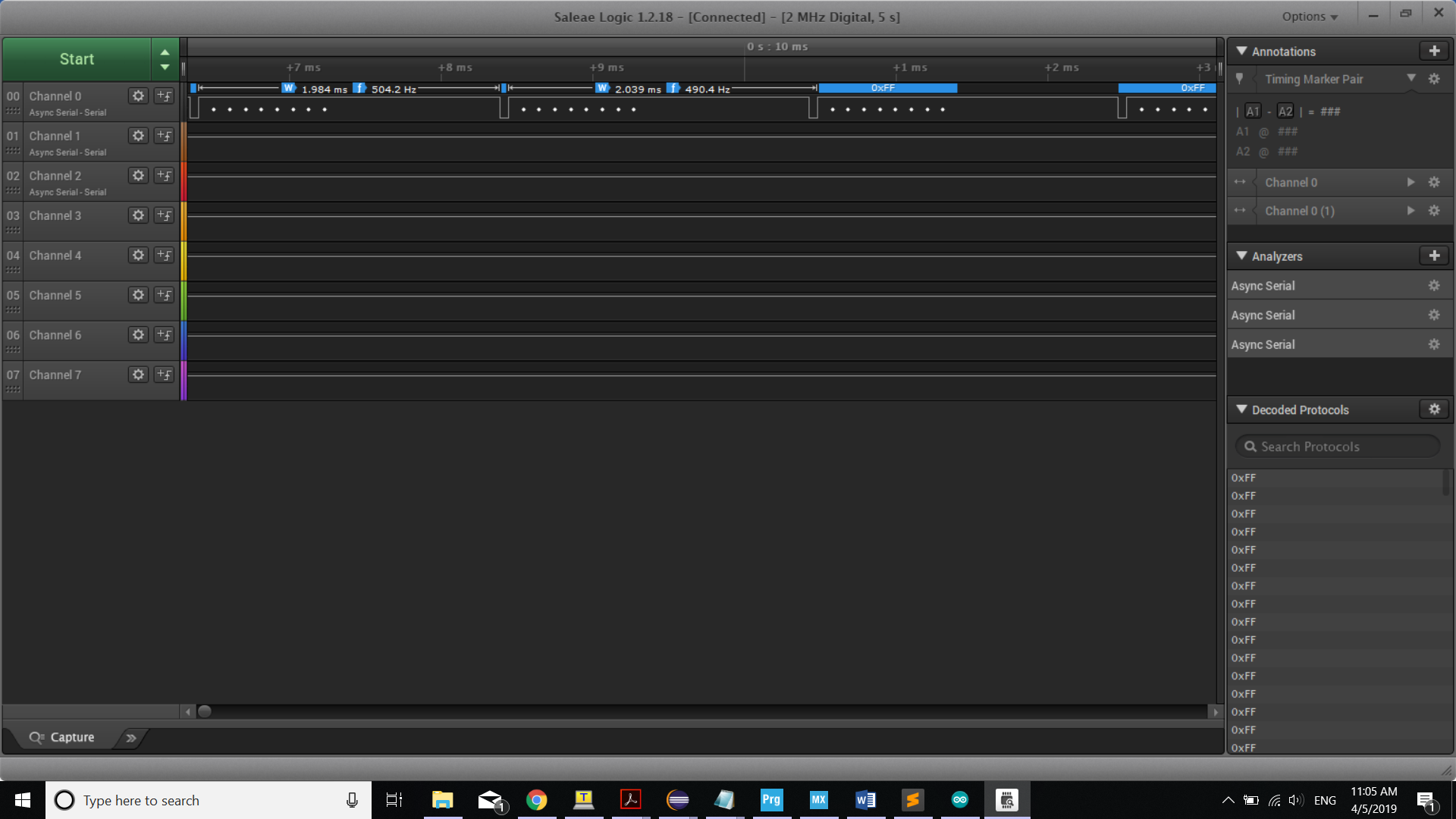
Serial.println(val2);

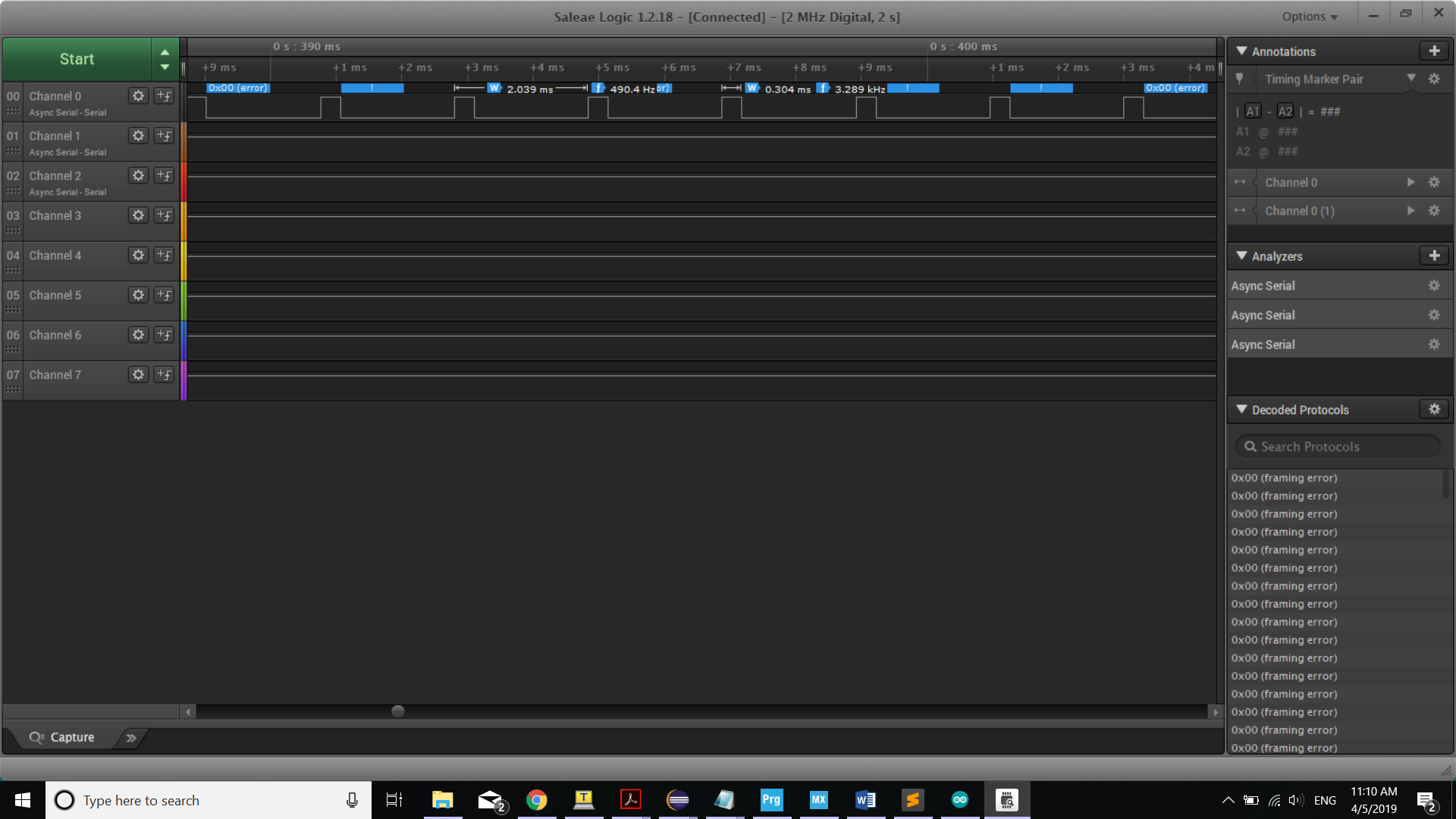
analogWrite(Pin1, val1);

analogWrite(Pin2,val2);

}

1. PWM outputs from the Arduino was monitored using Logic Analyzer.

When the value was set at 600 we got the above waveform with ON time of 0.704.

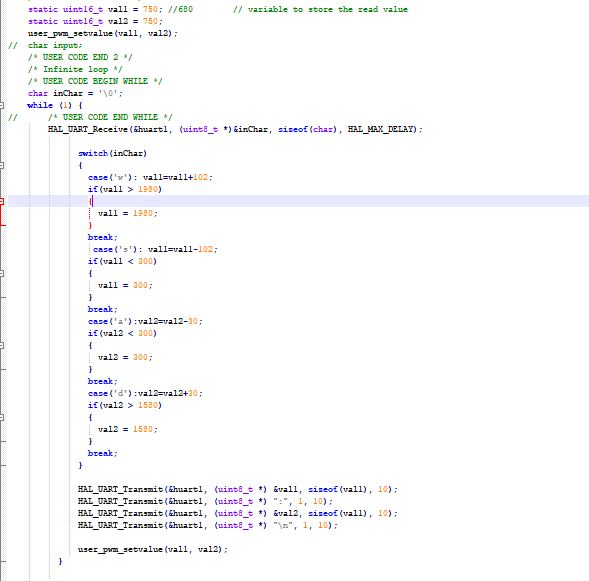
When the value was set at 760 we got the above waveform with ON time of 1.984.

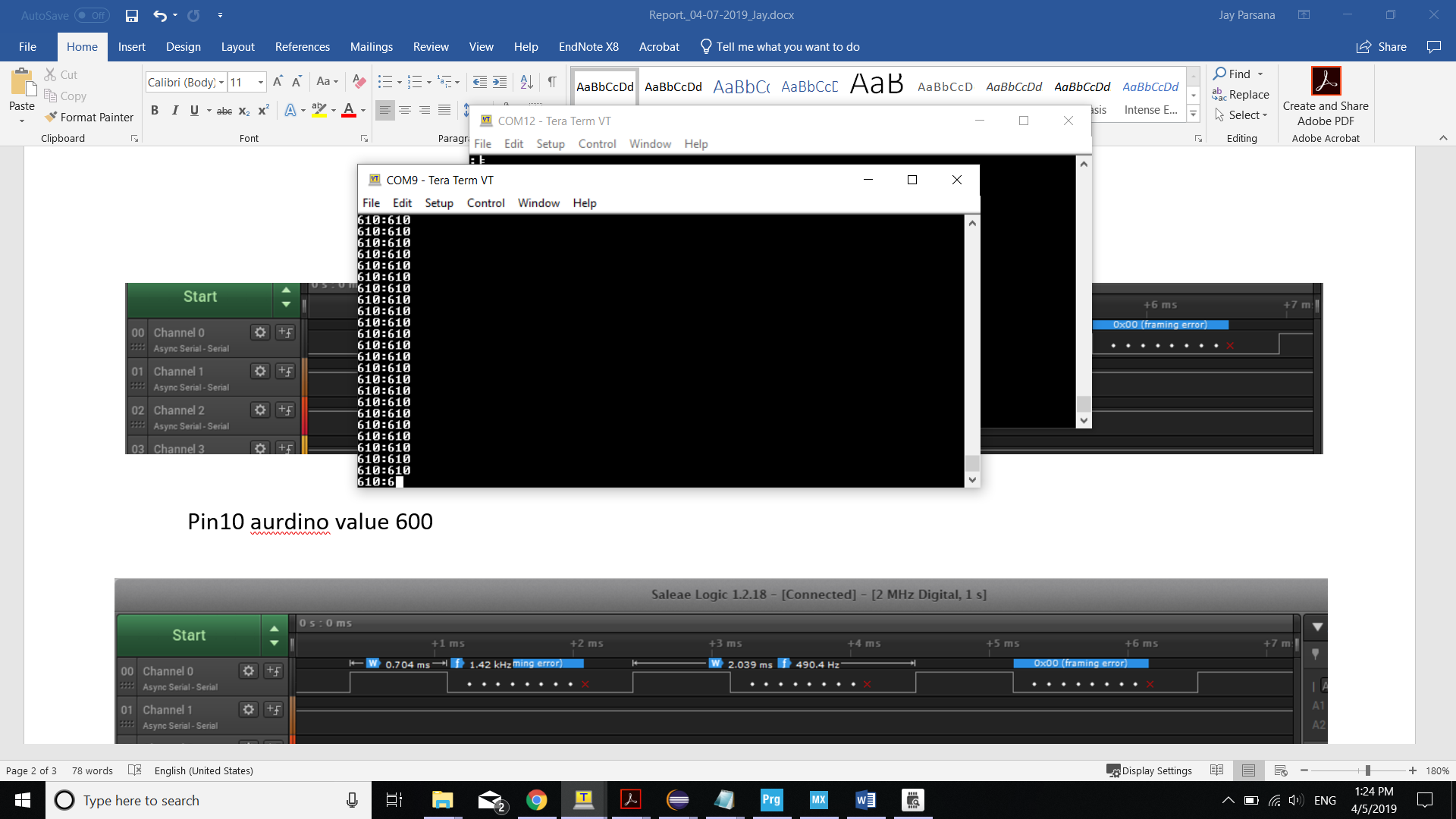
When the value was set at 760 we got the above waveform with ON time of 0.304.

1. So pressing ‘D’ increases val2, pressing ‘W’ increases val1, pressing ’A’ decreases val2, pressing ‘S; decreases val1 in the above Arduino code.
2. We ported this code into STM32 IoT Discovery Kit. We kept the prescaler to (80 – 1) and set the period to 2000. We analyzed the on-time for different values and thus after porting the values we varied the on period between 0.300ms to 2.00ms to achieve desired operation.
3. In uart stm code, uart\_set\_value() sets the value and value in decimal in logic analyser (0xff--> ON Time: 0.256ms (256us)). Using this logic, below table is computed showing the mapping of threshold values from Arduino code to STM code.

|  |  |
| --- | --- |
| **Arduino Threshold value** | **STM Discovery**  **(ON period in microseconds)** |
| 660 | 750 |
| 760 | 1980 |
| 550 | 300 |
| 710 | 1580 |

1. Below code snippet shows the implementation and second screenshot shows the output values for controlling the DC motor.





1. We were not able to run the wheelchair the first time, so we verified the connections several times and tried to move the wheelchair using remote control, but the chair didn’t move. Whenever we truned ON the remote of the wheelchair, it started beeping 5 times and did not move at all with the remote control. So using the manual we came to know that we need to shift the wheels mode between manual mode and electric mode.
2. Then we were able to move the wheelchair using remote. We again tried using STM board but it did not work. So we connected the Arduino again but still couldn’t move the wheelchair using the Arduino code. The remote started beeping with 7 beeps every time we started and connected it to Arduino. We came to know using manual that this is because of improper connections. But we verified the connections and after several trials and errors we found that the chair **PWM signals needs to be at neutral mode which is an ON time of 0.704ms and only then we need to start the remote.**
3. The wheelchair remote stopped beeping when the PWM signal is initially at 0.704ms for both the motors. Later on we changed the PWM ON period to move the chair using Arduino and later on tried the same thing on the STM32 IoT Discovery Kit and successfully moved the wheelchair.