**Software explanation**

|  |
| --- |
| **VARIABLES AND #INCLDUES** |
| #include <Wire.h>  #include <Adafruit\_GFX.h>  #include <Adafruit\_SSD1306.h>  #include <input.h>  #include <delays.h>  #define OLED\_RESET -1  TwoWire Wire2(PB7, PB6); //SDA SCL  Adafruit\_SSD1306 display(128, 64, &Wire2, OLED\_RESET);  int GPIO\_Bits[] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11};  //PROG  #define PROG\_GPIO PA12  #define ESC\_GPIO PB5  #define START\_GPIO PB8  #define STOP\_GPIO PA15  #define SET\_GPIO PB0  #define LEFT\_GPIO PB1  #define RIGHT\_GPIO PA0  #define UP\_GPIO PB3  #define DOWN\_GPIO PB4  INPUT\_T PROG\_Button(PROG\_GPIO, INPUT\_PULLUP, 20, LOW);  INPUT\_T ESC\_Button(ESC\_GPIO, INPUT\_PULLUP, 20, LOW);  INPUT\_T START\_Button(START\_GPIO, INPUT\_PULLUP, 20, LOW);  INPUT\_T STOP\_Button(STOP\_GPIO, INPUT\_PULLUP, 20, LOW);  INPUT\_T SET\_Button(SET\_GPIO, INPUT\_PULLUP, 20, LOW);  INPUT\_T LEFT\_Button(LEFT\_GPIO, INPUT\_PULLUP, 20, LOW);  INPUT\_T RIGHT\_Button(RIGHT\_GPIO, INPUT\_PULLUP, 20, LOW);  INPUT\_T UP\_Button(UP\_GPIO, INPUT\_PULLUP, 20, LOW);  INPUT\_T DOWN\_Button(DOWN\_GPIO, INPUT\_PULLUP, 20, LOW);  DELAYMS\_T PARAMETER\_CursorBlink\_DLY;  bool PARAMETER\_CursorBlinkState = true;  int Blink\_Rate = 500;  int MAX\_RPM\_Array [4] = {0, 0, 0, 0};  String StrMAX\_RPM\_Array[] = {"", "", "", ""};  String Concat\_StrMAX\_RPM\_Array;  int MIN\_RPM\_Array [4] = {0, 0, 0, 0};  String StrMIN\_RPM\_Array[] = {"", "", "", ""};  String Concat\_StrMIN\_RPM\_Array;  int Increase\_Array [3] = {0, 0, 0};  String StrIncrease\_Array[] = {"", "", ""};  String Concat\_StrIncrease\_Array;  int Decrease\_Array [3] = {0, 0, 0};  String StrDecrease\_Array[] = {"", "", ""};  String Concat\_StrDecrease\_Array;  int Static\_DecreaseArray [3] = {0, 0, 0};  int Static\_IncreaseArray [3] = {0, 0, 0};  int Static\_MIN\_RPM\_Array [4] = {0, 0, 0, 0};  int Static\_MAX\_RPM\_Array [4] = {0, 0, 0, 0};  int RPM = 0;  int DeviceParaPosition = 1;  int ParaArrowPosition[] = {5,};  int PROG\_Cursor\_Selection = 0; //0 = Operation Status | 1 = Device Parameter  int MAX\_Cursor\_Selection = 5;  int UI\_Navigation = 0;  int SystemStatus = 0;  /\*  UI\_NAVIGATION  0 = PROG UI  1 = OPERATION STATUS UI  2 = DEVICE PARAMETER UI  \*/  /\* DEVICE PARA NAVIGATION  MAX: 01 02 03 04 | 05  MIN: 06 07 08 09 | 10  INC: 11 12 13 | 14  DEC: 15 16 17 | 18  \*/ |
| **INSIDE SETUP()** |
| void setup() {  delay(100);  for (int i = 0; i < 12; i++)  {  pinMode(GPIO\_Bits[i], OUTPUT);  }  display.begin(SSD1306\_SWITCHCAPVCC, 0x3C);  SELECTION\_Prog\_UI(0);  } |
| **INSIDE LOOP()** |
| void loop()  {  Validation();  PROG\_ButtonFunction();  UP\_ButtonFunction();  DOWN\_ButtonFunction();  RIGHT\_ButtonFunction();  LEFT\_ButtonFunction();  SET\_ButtonFunction();  START\_ButtonFunction();  STOP\_ButtonFunction();  ESC\_ButtonFunction();  ConvertInt\_String();  } |
| **FUNCTIONS** |
| void OperationStatus(int x)  {  display.clearDisplay();  display.setTextSize(1);  display.setCursor(16, 0);  display.setTextColor(SSD1306\_WHITE);  display.println("Operation Status");  display.println();  display.println();  if (x == 0)  {  display.println("STATUS: STOPPED");  display.println();  }  else if (x == 1)  {  display.println("STATUS: RUNNING");  display.println();  }  display.println("RPM: " + String(RPM));  display.display();  }  void PROG\_UI()  {  display.clearDisplay();  display.setTextSize(2);  display.setTextColor(SSD1306\_WHITE);  display.setCursor(40, 0);  display.println("PROG");  display.setTextSize(1);  display.setCursor(0, 30);  display.println("Operation Status <-");  display.setCursor(0, 40);  display.println("Device Parameter ");  UI\_Navigation = 0;  PROG\_Cursor\_Selection = 0;  display.display();  }  void SELECTION\_Prog\_UI(int x)  {  if (x == 0)  {  PROG\_Cursor\_Selection = x;  Operation\_StatusCursor();  }  if (x == 1)  {  PROG\_Cursor\_Selection = x;  Device\_ParameterCursor();  }  }  void Operation\_StatusCursor()  {  display.clearDisplay();  display.setTextSize(2);  display.setTextColor(SSD1306\_WHITE);  display.setCursor(40, 0);  display.println("PROG");  display.setTextSize(1);  display.setCursor(0, 30);  display.println("Operation Status <-");  display.setCursor(0, 40);  display.println("Device Parameter ");  display.display();  }  void Device\_ParameterCursor()  {  display.clearDisplay();  display.setTextSize(2);  display.setTextColor(SSD1306\_WHITE);  display.setCursor(40, 0);  display.println("PROG");  display.setTextSize(1);  display.setCursor(0, 30);  display.println("Operation Status ");  display.setCursor(0, 40);  display.println("Device Parameter <-");  display.display();  }  void Parameter\_Cursor\_Blink(int x)  {  display.clearDisplay();  display.setTextSize(1);  display.setCursor(16, 0);  display.setTextColor(SSD1306\_WHITE);  display.println("Device Parameter");  display.setCursor(0, 15);  display.println("Current RPM: " + String(RPM));  if (PARAMETER\_CursorBlink\_DLY.dlyExpired())  {  if (PARAMETER\_CursorBlinkState == true)  {  PARAMETER\_CursorBlinkState = false;  PARAMETER\_CursorBlink\_DLY.dlySet(Blink\_Rate);  if (x < 5 || x == 5)  {  display.setCursor(0, 25);  display.println("MAX : " + String(Concat\_StrMAX\_RPM\_Array) + String(" <--"));  display.setCursor(0, 35);  display.println("MIN : " + String(Concat\_StrMIN\_RPM\_Array));  display.setCursor(0, 45);  display.println("Increase : " + String(Concat\_StrIncrease\_Array));  display.setCursor(0, 55);  display.println("Decrease : " + String(Concat\_StrDecrease\_Array));  display.display();  }  else if ((x > 5 && x < 10) || x == 10)  {  display.setCursor(0, 25);  display.println("MAX : " + String(Concat\_StrMAX\_RPM\_Array));  display.setCursor(0, 35);  display.println("MIN : " + String(Concat\_StrMIN\_RPM\_Array) + String(" <--"));  display.setCursor(0, 45);  display.println("Increase : " + String(Concat\_StrIncrease\_Array));  display.setCursor(0, 55);  display.println("Decrease : " + String(Concat\_StrDecrease\_Array));  display.display();  }  else if ((x > 10 && x < 14) || x == 14)  {  display.setCursor(0, 25);  display.println("MAX : " + String(Concat\_StrMAX\_RPM\_Array));  display.setCursor(0, 35);  display.println("MIN : " + String(Concat\_StrMIN\_RPM\_Array));  display.setCursor(0, 45);  display.println("Increase : " + String(Concat\_StrIncrease\_Array) + String(" <--"));  display.setCursor(0, 55);  display.println("Decrease : " + String(Concat\_StrDecrease\_Array));  display.display();  }  else if ((x > 14 && x < 18) || x == 18)  {  display.setCursor(0, 25);  display.println("MAX : " + String(Concat\_StrMAX\_RPM\_Array));  display.setCursor(0, 35);  display.println("MIN : " + String(Concat\_StrMIN\_RPM\_Array));  display.setCursor(0, 45);  display.println("Increase : " + String(Concat\_StrIncrease\_Array));  display.setCursor(0, 55);  display.println("Decrease : " + String(Concat\_StrDecrease\_Array) + String(" <--"));  display.display();  }  }  else  {  if (x == 1)  {  PARAMETER\_CursorBlinkState = true;  PARAMETER\_CursorBlink\_DLY.dlySet(Blink\_Rate);  display.setCursor(0, 25);  display.println("MAX : \_" + StrMAX\_RPM\_Array[1] + StrMAX\_RPM\_Array[2] + StrMAX\_RPM\_Array[3] + String(" <--"));  display.setCursor(0, 35);  display.println("MIN : " + String(Concat\_StrMIN\_RPM\_Array));  display.setCursor(0, 45);  display.println("Increase : " + String(Concat\_StrIncrease\_Array));  display.setCursor(0, 55);  display.println("Decrease : " + String(Concat\_StrDecrease\_Array));  display.display();  }  else if (x == 2)  {  PARAMETER\_CursorBlinkState = true;  PARAMETER\_CursorBlink\_DLY.dlySet(Blink\_Rate);  display.setCursor(0, 25);  display.println("MAX : " + StrMAX\_RPM\_Array[0] + String("\_") + StrMAX\_RPM\_Array[2] + StrMAX\_RPM\_Array[3] + String(" <--"));  display.setCursor(0, 35);  display.println("MIN : " + String(Concat\_StrMIN\_RPM\_Array));  display.setCursor(0, 45);  display.println("Increase : " + String(Concat\_StrIncrease\_Array));  display.setCursor(0, 55);  display.println("Decrease : " + String(Concat\_StrDecrease\_Array));  display.display();  }  else if (x == 3)  {  PARAMETER\_CursorBlinkState = true;  PARAMETER\_CursorBlink\_DLY.dlySet(Blink\_Rate);  display.setCursor(0, 25);  display.println("MAX : " + StrMAX\_RPM\_Array[0] + StrMAX\_RPM\_Array[1] + String("\_") + StrMAX\_RPM\_Array[3] + String(" <--"));  display.setCursor(0, 35);  display.println("MIN : " + String(Concat\_StrMIN\_RPM\_Array));  display.setCursor(0, 45);  display.println("Increase : " + String(Concat\_StrIncrease\_Array));  display.setCursor(0, 55);  display.println("Decrease : " + String(Concat\_StrDecrease\_Array));  display.display();  }  else if (x == 4)  {  PARAMETER\_CursorBlinkState = true;  PARAMETER\_CursorBlink\_DLY.dlySet(Blink\_Rate);  display.setCursor(0, 25);  display.println("MAX : " + StrMAX\_RPM\_Array[0] + StrMAX\_RPM\_Array[1] + StrMAX\_RPM\_Array[2] + String("\_") + String(" <--"));  display.setCursor(0, 35);  display.println("MIN : " + String(Concat\_StrMIN\_RPM\_Array));  display.setCursor(0, 45);  display.println("Increase : " + String(Concat\_StrIncrease\_Array));  display.setCursor(0, 55);  display.println("Decrease : " + String(Concat\_StrDecrease\_Array));  display.display();  }  else if (x == 6)  {  PARAMETER\_CursorBlinkState = true;  PARAMETER\_CursorBlink\_DLY.dlySet(Blink\_Rate);  display.setCursor(0, 25);  display.println("MAX : " + String(Concat\_StrMAX\_RPM\_Array));  display.setCursor(0, 35);  display.println("MIN : " + String("\_") + StrMIN\_RPM\_Array[1] + StrMIN\_RPM\_Array[2] + StrMIN\_RPM\_Array[3] + String(" <--"));  display.setCursor(0, 45);  display.println("Increase : " + String(Concat\_StrIncrease\_Array));  display.setCursor(0, 55);  display.println("Decrease : " + String(Concat\_StrDecrease\_Array));  display.display();  }  else if (x == 7)  {  PARAMETER\_CursorBlinkState = true;  PARAMETER\_CursorBlink\_DLY.dlySet(Blink\_Rate);  display.setCursor(0, 25);  display.println("MAX : " + String(Concat\_StrMAX\_RPM\_Array));  display.setCursor(0, 35);  display.println("MIN : " + StrMIN\_RPM\_Array[0] + String("\_") + StrMIN\_RPM\_Array[2] + StrMIN\_RPM\_Array[3] + String(" <--"));  display.setCursor(0, 45);  display.println("Increase : " + String(Concat\_StrIncrease\_Array));  display.setCursor(0, 55);  display.println("Decrease : " + String(Concat\_StrDecrease\_Array));  display.display();  }  else if (x == 8)  {  PARAMETER\_CursorBlinkState = true;  PARAMETER\_CursorBlink\_DLY.dlySet(Blink\_Rate);  display.setCursor(0, 25);  display.println("MAX : " + String(Concat\_StrMAX\_RPM\_Array));  display.setCursor(0, 35);  display.println("MIN : " + StrMIN\_RPM\_Array[0] + StrMIN\_RPM\_Array[1] + String("\_") + StrMIN\_RPM\_Array[3] + String(" <--"));  display.setCursor(0, 45);  display.println("Increase : " + String(Concat\_StrIncrease\_Array));  display.setCursor(0, 55);  display.println("Decrease : " + String(Concat\_StrDecrease\_Array));  display.display();  }  else if (x == 9)  {  PARAMETER\_CursorBlinkState = true;  PARAMETER\_CursorBlink\_DLY.dlySet(Blink\_Rate);  display.setCursor(0, 25);  display.println("MAX : " + String(Concat\_StrMAX\_RPM\_Array));  display.setCursor(0, 35);  display.println("MIN : " + StrMIN\_RPM\_Array[0] + StrMIN\_RPM\_Array[1] + StrMIN\_RPM\_Array[2] + String("\_") + String(" <--"));  display.setCursor(0, 45);  display.println("Increase : " + String(Concat\_StrIncrease\_Array));  display.setCursor(0, 55);  display.println("Decrease : " + String(Concat\_StrDecrease\_Array));  display.display();  }  else if (x == 11)  {  PARAMETER\_CursorBlinkState = true;  PARAMETER\_CursorBlink\_DLY.dlySet(Blink\_Rate);  display.setCursor(0, 25);  display.println("MAX : " + String(Concat\_StrMAX\_RPM\_Array));  display.setCursor(0, 35);  display.println("MIN : " + String(Concat\_StrMIN\_RPM\_Array));  display.setCursor(0, 45);  display.println("Increase : " + String("\_") + StrIncrease\_Array[1] + StrIncrease\_Array[2] + String(" <--"));  display.setCursor(0, 55);  display.println("Decrease : " + String(Concat\_StrDecrease\_Array));  display.display();  }  else if (x == 12)  {  PARAMETER\_CursorBlinkState = true;  PARAMETER\_CursorBlink\_DLY.dlySet(Blink\_Rate);  display.setCursor(0, 25);  display.println("MAX : " + String(Concat\_StrMAX\_RPM\_Array));  display.setCursor(0, 35);  display.println("MIN : " + String(Concat\_StrMIN\_RPM\_Array));  display.setCursor(0, 45);  display.println("Increase : " + StrIncrease\_Array[0] + String("\_") + StrIncrease\_Array[2] + String(" <--"));  display.setCursor(0, 55);  display.println("Decrease : " + String(Concat\_StrDecrease\_Array));  display.display();  }  else if (x == 13)  {  PARAMETER\_CursorBlinkState = true;  PARAMETER\_CursorBlink\_DLY.dlySet(Blink\_Rate);  display.setCursor(0, 25);  display.println("MAX : " + String(Concat\_StrMAX\_RPM\_Array));  display.setCursor(0, 35);  display.println("MIN : " + String(Concat\_StrMIN\_RPM\_Array));  display.setCursor(0, 45);  display.println("Increase : " + StrIncrease\_Array[0] + StrIncrease\_Array[1] + String("\_") + String(" <--"));  display.setCursor(0, 55);  display.println("Decrease : " + String(Concat\_StrDecrease\_Array));  display.display();  }  else if (x == 15)  {  PARAMETER\_CursorBlinkState = true;  PARAMETER\_CursorBlink\_DLY.dlySet(Blink\_Rate);  display.setCursor(0, 25);  display.println("MAX : " + String(Concat\_StrMAX\_RPM\_Array));  display.setCursor(0, 35);  display.println("MIN : " + String(Concat\_StrMIN\_RPM\_Array));  display.setCursor(0, 45);  display.println("Increase : " + String(Concat\_StrIncrease\_Array));  display.setCursor(0, 55);  display.println("Decrease : " + String("\_") + StrDecrease\_Array[1] + StrDecrease\_Array[2] + String(" <--"));  display.display();  }  else if (x == 16)  {  PARAMETER\_CursorBlinkState = true;  PARAMETER\_CursorBlink\_DLY.dlySet(Blink\_Rate);  display.setCursor(0, 25);  display.println("MAX : " + String(Concat\_StrMAX\_RPM\_Array));  display.setCursor(0, 35);  display.println("MIN : " + String(Concat\_StrMIN\_RPM\_Array));  display.setCursor(0, 45);  display.println("Increase : " + String(Concat\_StrIncrease\_Array));  display.setCursor(0, 55);  display.println("Decrease : " + StrDecrease\_Array[0] + String("\_") + StrDecrease\_Array[2] + String(" <--"));  display.display();  }  else if (x == 17)  {  PARAMETER\_CursorBlinkState = true;  PARAMETER\_CursorBlink\_DLY.dlySet(Blink\_Rate);  display.setCursor(0, 25);  display.println("MAX : " + String(Concat\_StrMAX\_RPM\_Array));  display.setCursor(0, 35);  display.println("MIN : " + String(Concat\_StrMIN\_RPM\_Array));  display.setCursor(0, 45);  display.println("Increase : " + String(Concat\_StrIncrease\_Array));  display.setCursor(0, 55);  display.println("Decrease : " + StrDecrease\_Array[0] + StrDecrease\_Array[1] + String("\_") + String(" <--"));  display.display();  }  if (x == 5 || x == 10 || x == 14 || x == 18)  {  PARAMETER\_CursorBlinkState = true;  PARAMETER\_CursorBlink\_DLY.dlySet(Blink\_Rate);  display.setCursor(0, 25);  display.println("MAX : " + String(Concat\_StrMAX\_RPM\_Array));  display.setCursor(0, 35);  display.println("MIN : " + String(Concat\_StrMIN\_RPM\_Array));  display.setCursor(0, 45);  display.println("Increase : " + String(Concat\_StrIncrease\_Array));  display.setCursor(0, 55);  display.println("Decrease : " + String(Concat\_StrDecrease\_Array));  display.display();  }  }  }  }  void ConvertValues(int x)  {  display.clearDisplay();  display.setTextSize(1);  display.setCursor(16, 0);  display.setTextColor(SSD1306\_WHITE);  display.println("Device Parameter");  for (int x = 0; x < 5; x++)  {  if (x < 3)  {  StrMAX\_RPM\_Array[x] = String(MAX\_RPM\_Array[x]);  StrMIN\_RPM\_Array[x] = String(MIN\_RPM\_Array[x]);  StrIncrease\_Array[x] = String(Increase\_Array[x]);  StrDecrease\_Array[x] = String(Decrease\_Array[x]);  }  else if (x == 3 )  {  StrMAX\_RPM\_Array[x] = String(MAX\_RPM\_Array[x]);  StrMIN\_RPM\_Array[x] = String(MIN\_RPM\_Array[x]);  }  }  Concat\_StrMAX\_RPM\_Array = StrMAX\_RPM\_Array[0] + StrMAX\_RPM\_Array[1] + StrMAX\_RPM\_Array[2] + StrMAX\_RPM\_Array[3];  Concat\_StrMIN\_RPM\_Array = StrMIN\_RPM\_Array[0] + StrMIN\_RPM\_Array[1] + StrMIN\_RPM\_Array[2] + StrMIN\_RPM\_Array[3];  Concat\_StrIncrease\_Array = StrIncrease\_Array[0] + StrIncrease\_Array[1] + StrIncrease\_Array[2];  Concat\_StrDecrease\_Array = StrDecrease\_Array[0] + StrDecrease\_Array[1] + StrDecrease\_Array[2];  display.setCursor(0, 15);  Parameter\_Cursor\_Blink(x);  }  void DevicePara()  {  display.setCursor(0, 25);  display.println("MAX : " + String(Concat\_StrMAX\_RPM\_Array));  display.setCursor(0, 35);  display.println("MIN : " + String(Concat\_StrMIN\_RPM\_Array));  display.setCursor(0, 45);  display.println("Increase : " + String(Concat\_StrIncrease\_Array));  display.setCursor(0, 55);  display.println("Decrease : " + String(Concat\_StrDecrease\_Array));  //display.display();  }  void Increase\_MAX()  {  if (DeviceParaPosition == 1)  {  if (MAX\_RPM\_Array[0] == 1)  {  MAX\_RPM\_Array[0] = 1;  }  else  {  MAX\_RPM\_Array[0] = MAX\_RPM\_Array[0] + 1;  }  }  if (DeviceParaPosition == 2)  {  if (MAX\_RPM\_Array[1] == 9)  {  MAX\_RPM\_Array [1] = 9;  }  else  {  MAX\_RPM\_Array[1] = MAX\_RPM\_Array[1] + 1;  }  }  if (DeviceParaPosition == 3)  {  if (MAX\_RPM\_Array[2] == 9)  {  MAX\_RPM\_Array [2] = 9;  }  else  {  MAX\_RPM\_Array[2] = MAX\_RPM\_Array[2] + 1;  }  }  if (DeviceParaPosition == 4)  {  if (MAX\_RPM\_Array[3] == 9)  {  MAX\_RPM\_Array [3] = 9;  }  else  {  MAX\_RPM\_Array[3] = MAX\_RPM\_Array[3] + 1;  }  }  }  void Decrease\_MAX()  {  if (DeviceParaPosition == 1)  {  if (MAX\_RPM\_Array[0] == 0)  {  MAX\_RPM\_Array[0] = 0;  }  else  {  MAX\_RPM\_Array[0] = MAX\_RPM\_Array[0] - 1;  }  }  if (DeviceParaPosition == 2)  {  if (MAX\_RPM\_Array[1] == 0)  {  MAX\_RPM\_Array [1] = 0;  }  else  {  MAX\_RPM\_Array[1] = MAX\_RPM\_Array[1] - 1;  }  }  if (DeviceParaPosition == 3)  {  if (MAX\_RPM\_Array[2] == 0)  {  MAX\_RPM\_Array [2] = 0;  }  else  {  MAX\_RPM\_Array[2] = MAX\_RPM\_Array[2] - 1;  }  }  if (DeviceParaPosition == 4)  {  if (MAX\_RPM\_Array[3] == 0)  {  MAX\_RPM\_Array [3] = 0;  }  else  {  MAX\_RPM\_Array[3] = MAX\_RPM\_Array[3] - 1;  }  }  }  void Increase\_MIN()  {  if (DeviceParaPosition == 6)  {  if (MIN\_RPM\_Array[0] == 1)  {  MIN\_RPM\_Array[0] = 1;  }  else  {  MIN\_RPM\_Array[0] = MIN\_RPM\_Array[0] + 1;  }  }  if (DeviceParaPosition == 7)  {  if (MIN\_RPM\_Array[1] == 9)  {  MIN\_RPM\_Array [1] = 9;  }  else  {  MIN\_RPM\_Array[1] = MIN\_RPM\_Array[1] + 1;  }  }  if (DeviceParaPosition == 8)  {  if (MIN\_RPM\_Array[2] == 9)  {  MIN\_RPM\_Array [2] = 9;  }  else  {  MIN\_RPM\_Array[2] = MIN\_RPM\_Array[2] + 1;  }  }  if (DeviceParaPosition == 9)  {  if (MIN\_RPM\_Array[3] == 9)  {  MIN\_RPM\_Array [3] = 9;  }  else  {  MIN\_RPM\_Array[3] = MIN\_RPM\_Array[3] + 1;  }  }  }  void Decrease\_MIN()  {  if (DeviceParaPosition == 6)  {  if (MIN\_RPM\_Array[0] == 0)  {  MIN\_RPM\_Array[0] = 0;  }  else  {  MIN\_RPM\_Array[0] = MIN\_RPM\_Array[0] - 1;  }  }  if (DeviceParaPosition == 7)  {  if (MIN\_RPM\_Array[1] == 0)  {  MIN\_RPM\_Array [1] = 0;  }  else  {  MIN\_RPM\_Array[1] = MIN\_RPM\_Array[1] - 1;  }  }  if (DeviceParaPosition == 8)  {  if (MIN\_RPM\_Array[2] == 0)  {  MIN\_RPM\_Array [2] = 0;  }  else  {  MIN\_RPM\_Array[2] = MIN\_RPM\_Array[2] - 1;  }  }  if (DeviceParaPosition == 9)  {  if (MIN\_RPM\_Array[3] == 0)  {  MIN\_RPM\_Array [3] = 0;  }  else  {  MIN\_RPM\_Array[3] = MIN\_RPM\_Array[3] - 1;  }  }  }  void Increase\_Step\_UP()  {  if (DeviceParaPosition == 11)  {  if (Increase\_Array[0] == 9)  {  Increase\_Array[0] = 9;  }  else  {  Increase\_Array[0] = Increase\_Array[0] + 1;  }  }  if (DeviceParaPosition == 12)  {  if (Increase\_Array[1] == 9)  {  Increase\_Array [1] = 9;  }  else  {  Increase\_Array[1] = Increase\_Array[1] + 1;  }  }  if (DeviceParaPosition == 13)  {  if (Increase\_Array[2] == 9)  {  Increase\_Array [2] = 9;  }  else  {  Increase\_Array[2] = Increase\_Array[2] + 1;  }  }  }  void Increase\_Step\_DOWN()  {  if (DeviceParaPosition == 11)  {  if (Increase\_Array[0] == 0)  {  Increase\_Array[0] = 0;  }  else  {  Increase\_Array[0] = Increase\_Array[0] - 1;  }  }  if (DeviceParaPosition == 12)  {  if (Increase\_Array[1] == 0)  {  Increase\_Array [1] = 0;  }  else  {  Increase\_Array[1] = Increase\_Array[1] - 1;  }  }  if (DeviceParaPosition == 13)  {  if (Increase\_Array[2] == 0)  {  Increase\_Array [2] = 0;  }  else  {  Increase\_Array[2] = Increase\_Array[2] - 1;  }  }  }  void Decrease\_Step\_UP()  {  if (DeviceParaPosition == 15)  {  if (Decrease\_Array[0] == 9)  {  Decrease\_Array[0] = 9;  }  else  {  Decrease\_Array[0] = Decrease\_Array[0] + 1;  }  }  if (DeviceParaPosition == 16)  {  if (Decrease\_Array[1] == 9)  {  Decrease\_Array [1] = 9;  }  else  {  Decrease\_Array[1] = Decrease\_Array[1] + 1;  }  }  if (DeviceParaPosition == 17)  {  if (Decrease\_Array[2] == 9)  {  Decrease\_Array [2] = 9;  }  else  {  Decrease\_Array[2] = Decrease\_Array[2] + 1;  }  }  }  void Decrease\_Step\_DOWN()  {  if (DeviceParaPosition == 15)  {  if (Decrease\_Array[0] == 0)  {  Decrease\_Array[0] = 0;  }  else  {  Decrease\_Array[0] = Decrease\_Array[0] - 1;  }  }  if (DeviceParaPosition == 16)  {  if (Decrease\_Array[1] == 0)  {  Decrease\_Array [1] = 0;  }  else  {  Decrease\_Array[1] = Decrease\_Array[1] - 1;  }  }  if (DeviceParaPosition == 17)  {  if (Decrease\_Array[2] == 0)  {  Decrease\_Array [2] = 0;  }  else  {  Decrease\_Array[2] = Decrease\_Array[2] - 1;  }  }  }  void Validation()  {  if (UI\_Navigation == 2 && Concat\_StrMAX\_RPM\_Array.toInt() > 1500)  {  MAX\_RPM\_Array[0] = 1;  MAX\_RPM\_Array[1] = 5;  MAX\_RPM\_Array[2] = 0;  MAX\_RPM\_Array[3] = 0;  }  if (UI\_Navigation == 2 && Concat\_StrMAX\_RPM\_Array.toInt() < Concat\_StrMIN\_RPM\_Array.toInt())  {  MIN\_RPM\_Array[0] = 0;  MIN\_RPM\_Array[1] = 0;  MIN\_RPM\_Array[2] = 0;  MIN\_RPM\_Array[3] = 0;  }  if (UI\_Navigation == 2 && Concat\_StrIncrease\_Array.toInt() > 999)  {  Increase\_Array[0] = 0;  Increase\_Array[1] = 0;  Increase\_Array[2] = 0;  }  if (UI\_Navigation == 2 && Concat\_StrDecrease\_Array.toInt() > 999)  {  Decrease\_Array[0] = 0;  Decrease\_Array[1] = 0;  Decrease\_Array[2] = 0;  }  }  void Set\_Parameter()  {  for (int x = 0; x < 5; x++)  {  if (x < 3)  {  Static\_MAX\_RPM\_Array[x] = MAX\_RPM\_Array[x];  Static\_MIN\_RPM\_Array[x] = MIN\_RPM\_Array[x];  Static\_IncreaseArray[x] = Increase\_Array[x];  Static\_DecreaseArray[x] = Decrease\_Array[x];  }  else if (x == 3 )  {  Static\_MAX\_RPM\_Array[x] = MAX\_RPM\_Array[x];  Static\_MIN\_RPM\_Array[x] = MIN\_RPM\_Array[x];  }  }  RPM = Concat\_StrMIN\_RPM\_Array.toInt();  SystemStatus = 1;  }  void Increase\_STEP()  {  RPM = RPM + Concat\_StrIncrease\_Array.toInt();  if (RPM > Concat\_StrMAX\_RPM\_Array.toInt())  {  RPM = Concat\_StrMAX\_RPM\_Array.toInt();  }  }  void Decrease\_STEP()  {  RPM = RPM - Concat\_StrDecrease\_Array.toInt();  if (RPM < Concat\_StrMIN\_RPM\_Array.toInt())  {  RPM = Concat\_StrMIN\_RPM\_Array.toInt();  }  }  void RevertValues()  {  for (int x = 0; x < 5; x++)  {  if (x < 3)  {  MAX\_RPM\_Array[x] = Static\_MAX\_RPM\_Array[x];  MIN\_RPM\_Array[x] = Static\_MIN\_RPM\_Array[x];  Increase\_Array[x] = Static\_IncreaseArray[x];  Decrease\_Array[x] = Static\_DecreaseArray[x];  }  else if (x == 3 )  {  MAX\_RPM\_Array[x] = Static\_MAX\_RPM\_Array[x];  MIN\_RPM\_Array[x] = Static\_MIN\_RPM\_Array[x];  }  }  }  void Output\_BITS(int mapped\_RPM\_Resolution)  {  for (int i = 0; i < 11; i++)  {  if (bitRead(mapped\_RPM\_Resolution, i) == 1)  {  digitalWrite(GPIO\_Bits[i], HIGH);  }  else  {  digitalWrite(GPIO\_Bits[i], LOW);  }  }  }  ConvertInt\_String()  {  if (UI\_Navigation == 2)  {  ConvertValues(DeviceParaPosition);  }  }  PROG\_ButtonFunction()  {  if (PROG\_Button.justPressed())  {  PROG\_UI();  }  }  ESC\_ButtonFunction()  {  if (ESC\_Button.justPressed())  {  if (UI\_Navigation == 2)  {  UI\_Navigation = 1;  RevertValues();  OperationStatus(SystemStatus);  }  }  }  STOP\_ButtonFunction()  {  if (STOP\_Button.justPressed())  {  if (UI\_Navigation == 1)  {  SystemStatus = 0;  OperationStatus(SystemStatus);  Output\_BITS(0);  }  }  }  START\_ButtonFunction()  {  if (START\_Button.justPressed())  {  if (UI\_Navigation == 1)  {  SystemStatus = 1;  OperationStatus(SystemStatus);  Output\_BITS(map(RPM,0,1500,0,2047));  }  }  }  SET\_ButtonFunction()  {  if (SET\_Button.justPressed())  {  if (UI\_Navigation == 0 && PROG\_Cursor\_Selection == 0 ) //PROG MENU, Operation Status Selected  {  UI\_Navigation = 1;  OperationStatus(SystemStatus);  }  else if (UI\_Navigation == 0 && PROG\_Cursor\_Selection == 1 ) //PROG MENU, Device Parameter Selected  {  UI\_Navigation = 2;  DevicePara();  }  else if (UI\_Navigation == 2)  {  UI\_Navigation = 1;  Set\_Parameter();  SystemStatus = 1;  Output\_BITS(map(RPM,0,1500,0,2047));  OperationStatus(SystemStatus);  }  }  }  LEFT\_ButtonFunction()  {  if (LEFT\_Button.justPressed())  {  if (UI\_Navigation == 2)  {  if (DeviceParaPosition != 1)  {  DeviceParaPosition = DeviceParaPosition - 1;  }  else if (DeviceParaPosition == 1)  {  DeviceParaPosition = 1;  }  }  }  }  RIGHT\_ButtonFunction()  {  if (RIGHT\_Button.justPressed())  {  if (UI\_Navigation == 2)  {  if (DeviceParaPosition != 18)  {  DeviceParaPosition = DeviceParaPosition + 1;  }  else if (DeviceParaPosition == 18)  {  DeviceParaPosition = 18;  }  }  }  }  DOWN\_ButtonFunction()  {  if (DOWN\_Button.justPressed())  {  if (UI\_Navigation == 0)  {  SELECTION\_Prog\_UI(1);  }  if (UI\_Navigation == 2)  {  Decrease\_MAX();  Decrease\_MIN();  Increase\_Step\_DOWN();  Decrease\_Step\_DOWN();  }  if (UI\_Navigation == 2 && DeviceParaPosition == 5)  {  DeviceParaPosition = 10;  }  else if (UI\_Navigation == 2 && DeviceParaPosition == 10)  {  DeviceParaPosition = 14;  }  else if (UI\_Navigation == 2 && DeviceParaPosition == 14)  {  DeviceParaPosition = 18;  }  if (UI\_Navigation == 1 && SystemStatus == 1)  {  Decrease\_STEP();  OperationStatus(SystemStatus);  Output\_BITS(map(RPM,0,1500,0,2047));  }  }  }  UP\_ButtonFunction()  {  if (UP\_Button.justPressed())  {  if (UI\_Navigation == 0)  {  SELECTION\_Prog\_UI(0);  }  if (UI\_Navigation == 2)  {  Increase\_MAX();  Increase\_MIN();  Increase\_Step\_UP();  Decrease\_Step\_UP();  } //END OF UI NAVIGATION = 2 IF STATEMENT  if (UI\_Navigation == 2 && DeviceParaPosition == 10)  {  DeviceParaPosition = 5;  }  if (UI\_Navigation == 2 && DeviceParaPosition == 14)  {  DeviceParaPosition = 10;  }  if (UI\_Navigation == 2 && DeviceParaPosition == 18)  {  DeviceParaPosition = 14;  }  if (UI\_Navigation == 1 && SystemStatus == 1)  {  Increase\_STEP();  OperationStatus(SystemStatus);  //OperationStatus\_UI(SystemStatus)  Output\_BITS(map(RPM,0,1500,0,2047));  }  }  } |

|  |  |
| --- | --- |
| **Validation FUNCTION** | |
| Validation();  void Validation()  {  if (UI\_Navigation == 2 && Concat\_StrMAX\_RPM\_Array.toInt() > 1500)  {  MAX\_RPM\_Array[0] = 1;  MAX\_RPM\_Array[1] = 5;  MAX\_RPM\_Array[2] = 0;  MAX\_RPM\_Array[3] = 0;  }  if (UI\_Navigation == 2 && Concat\_StrMAX\_RPM\_Array.toInt() < Concat\_StrMIN\_RPM\_Array.toInt())  {  MIN\_RPM\_Array[0] = 0;  MIN\_RPM\_Array[1] = 0;  MIN\_RPM\_Array[2] = 0;  MIN\_RPM\_Array[3] = 0;  }  if (UI\_Navigation == 2 && Concat\_StrIncrease\_Array.toInt() > 999)  {  Increase\_Array[0] = 0;  Increase\_Array[1] = 0;  Increase\_Array[2] = 0;  }  if (UI\_Navigation == 2 && Concat\_StrDecrease\_Array.toInt() > 999)  {  Decrease\_Array[0] = 0;  Decrease\_Array[1] = 0;  Decrease\_Array[2] = 0;  }  } | |
| Explanation | This function make sure that   * MAX\_RPM does not exceeds 1500. * Sets MIN\_RPM to 0 if MAX\_RPM is less than MIN\_RPM. * If the Increase or Decrease exceeds 999 it sets the values to 0.   This function will only be executed if the user is in Device Parameter UI. |

|  |  |
| --- | --- |
| **PROG BUTTON FUNCTION** | |
| PROG\_ButtonFunction();  void PROG\_ButtonFunction()  {  if (PROG\_Button.justPressed())  {  PROG\_UI();  }  }  void PROG\_UI()  {  display.clearDisplay();  display.setTextSize(2);  display.setTextColor(SSD1306\_WHITE);  display.setCursor(40, 0);  display.println("PROG");  display.setTextSize(1);  display.setCursor(0, 30);  display.println("Operation Status <-");  display.setCursor(0, 40);  display.println("Device Parameter ");  UI\_Navigation = 0;  PROG\_Cursor\_Selection = 0;  display.display();  } | |
| Explanation | The PROG\_ButtonFunction has another function inside it. Whenever the PROG Button is pressed, it will call another function which is “PROG\_UI();”.  The PROG\_UI function will   * Display how the page for PROG UI which consists of “PROG” as the header below the header would be “Operation Status” and “Device Parameter”. * After it displays, it will set the variable UI\_Navigation to 0. This is to indicate that the user is in PROG UI. * It will also set the PROG\_Cursor\_Selection to 0. Which will result in Operation Status being selected as the default. |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **UP BUTTON FUNCTION** | | | | | | | |
| UP\_ButtonFunction();  void UP\_ButtonFunction()  {  if (UP\_Button.justPressed())  {  if (UI\_Navigation == 0)  {  SELECTION\_Prog\_UI(0);  }  if (UI\_Navigation == 2)  {  Increase\_MAX();  Increase\_MIN();  Increase\_Step\_UP();  Decrease\_Step\_UP();  }  if (UI\_Navigation == 2 && DeviceParaPosition == 10)  {  DeviceParaPosition = 5;  }  if (UI\_Navigation == 2 && DeviceParaPosition == 14)  {  DeviceParaPosition = 10;  }  if (UI\_Navigation == 2 && DeviceParaPosition == 18)  {  DeviceParaPosition = 14;  }  if (UI\_Navigation == 1 && SystemStatus == 1)  {  Increase\_STEP();  OperationStatus(SystemStatus);  Output\_BITS(map(RPM,0,1500,0,2047));  }  }  } | | | | | | | |
| Explanation | **UI\_Navigation = 0** | | | | | | |
| The UP\_ButtonFunction is used across all different interfaces therefore the variable UI\_Navigation is to keep track of which UI the user is currently at.  If the variable of the UI\_Navigation = 0, this indicates that the user is in the PROG UI so then it will execute the if statement which have the condition ”UI\_Navigation == 0”. After that it will the function “SELECTION\_Prog\_UI(0);” This function selects the Operation Status. | | | | | | |
| **UI\_Navigation = 1** | | | | | | |
| If the variable of the UI\_Navigation = 1, this indicates that the user is in the Operation Status UI. In order for user to use the UP BUTTON in the Operation Status UI, the condition of UI\_Navigation == 1 and SystemStatus == 1 must be met. If the condition is met the user will be able to increase the STEP which increases the RPM. When the user press the UP Button in Operation Status UI, it calls 3 sub functions.   * Increase\_STEP(); * OperationStatus(SystemStatus); * Output\_BITS(map(RPM,0,1500,0,2047));   Increase\_STEP();   * This function increases the RPM value by adding the value from the variable “Concat\_StrIncrease\_Array” * If the value of RPM were to exceed the MAX\_RPM value, RPM value will be the user defined MAX\_RPM.   OperationStatus(SystemStatus);   * This function takes an integer value to determine if the System is running or stopped. * If the variable SystemStatus = 0, it will display “STATUS: STOPPED” followed by “RPM: rpmValue” else if variable SystemStatus == 1 then it will display “STATUS: RUNNING” followed by “RPM: rpmValue“.   Output\_BITS(map(RPM,0,1500,0,2047));   * This function sets the 11 GPIO HIGH or LOW which are responsible from Bit 1 to Bit 11. * The map(RPM,0,1500,0,2047) means that the value RPM will be mapped with the 2047 levels that the 11 bit R2R DAC offers. * 0RPM will be mapped to LEVEL 0 * 15000RPM will be mapped to LEVEL 2047 * Lets say the user were to set the RPM to be 1 so the mapped level would be 1. So then the Output\_BITS(); Function will then take the mapped level of 1. From the integer 1 will be converted to binary so in binary format it would be 000 0000 0001. Inside the Output\_BITS() is a for loop which reads the input in binary format which would read from LSB to MSB. So when it goes through and if the bit is a 0, it will sets the GPIO\_Bits[i] which is responsible for that bit a LOW if it’s a 1 then it will output a HIGH. | | | | | | |
| **UI\_Navigation = 2** | | | | | | |
| In order to explain function under the condition of UI\_Navigation == 2, you will have to understand the variable DeviceParaPosition. Below is the layout of the Device Parameter UI. | | | | | | |
| **Device Parameter** | | | | | | |
| **Current RPM** | | **0000** | | | | |
| **MAX** | | **01** | **02** | **03** | **04** | **05** |
| **MIN** | | **06** | **07** | **08** | **09** | **10** |
| **Increase** | | **11** | **12** | **13** |  | **14** |
| **Decrease** | | **15** | **16** | **17** |  | **18** |
| The cells that are highlighted in green are position of each individual values in an integer.  Assume that   * MAX RPM row is set to 1423. * MIN RPM row is set to 0500. * Increase row is set to 020. * Decrease row is set to 010. | | | | | | |
| **Place Value**  **MAX & MIN** | **Thousands** | | **Hundreds** | **Tens** | **Ones** | **Cursor** |
| **MAX** | **1** | | **4** | **2** | **3** | **<----** |
| **MIN** | **0** | | **5** | **0** | **0** |  |
| **Increase** | **0** | | **2** | **0** |  |  |
| **Decrease** | **0** | | **1** | **0** |  |  |
| **Place Value**  **Increase & Decrease** | **Hundreds** | | **Tens** | **Ones** |  | **Cursor** |
| MAX Thousands Place is represented by 1 in DeviceParaPosition. MAX Hundreds Place is represented by 2 in DeviceParaPosition.  MAX Tens Place is represented by 3 in DeviceParaPosition. MAX Ones Place is represented by 4 in DeviceParaPosition.  Cursor which indicates the current selected ROW for MAX is represented by 5 in DeviceParaPosition.  MIN Thousands Place is represented by 6 in DeviceParaPosition.  MIN Hundreds Place is represented by 7 in DeviceParaPosition.  MIN Tens Place is represented by 8 in DeviceParaPosition.  MIN Ones Place is represented by 9 in DeviceParaPosition.  Cursor which indicates the current selected ROW for MIN is represented by 10 in DeviceParaPosition.  However, for the Increase row. It is different from MAX and MIN ROW.  Increase Hundreds is represented by 11 in DeviceParaPosition. Increase Tens is represented by 12 in DeviceParaPosition.  Increase Ones is represented by 13 in DeviceParaPosition.  Cursor which indicates the current selected ROW for Increase is represented by 14 in DeviceParaPosition.  Decrease Hundreds Place is represented by 15 in DeviceParaPosition.  Decrease Tens Place is represented by 16 in DeviceParaPosition.  Decrease Ones Place is represented by 17 in DeviceParaPosition.  Cursor which indicates the current selected ROW for Decrease is represented by 18 in DeviceParaPosition.  In summary, the DeviceParaPosition variable keep tracks of which number the user is currently trying to configure. | | | | | | |
| The first if statement with the condition “**if(UI\_Navigation == 2)**” is responsible for changing the values of number in MAX,MIN,Increase and Decrease row. Inside there, there are 4 subfunctions.   * Increase\_MAX(); * Increase\_MIN(); * Increase\_Step\_UP(); * Decrease\_Step\_UP();   These 4 Functions takes the DeviceParaPosition and change the values accordingly. | | | | | | |
| The are other if statement with extra conditions which is  **“if (UI\_Navigation == 2 && DeviceParaPosition == 10)”**  This 2 condition will allow us to move the Cursor to the ROW above the current selected ROW. | | | | | | |

|  |  |
| --- | --- |
| **DOWN BUTTON FUNCTION** | |
| DOWN\_ButtonFunction();  DOWN\_ButtonFunction()  {  if (DOWN\_Button.justPressed())  {  if (UI\_Navigation == 0)  {  SELECTION\_Prog\_UI(1);  }  if (UI\_Navigation == 2)  {  Decrease\_MAX();  Decrease\_MIN();  Increase\_Step\_DOWN();  Decrease\_Step\_DOWN();  }  if (UI\_Navigation == 2 && DeviceParaPosition == 5)  {  DeviceParaPosition = 10;  }  else if (UI\_Navigation == 2 && DeviceParaPosition == 10)  {  DeviceParaPosition = 14;  }  else if (UI\_Navigation == 2 && DeviceParaPosition == 14)  {  DeviceParaPosition = 18;  }  if (UI\_Navigation == 1 && SystemStatus == 1)  {  Decrease\_STEP();  OperationStatus(SystemStatus);  Output\_BITS(map(RPM,0,1500,0,2047));  }  }  } | |
| Explanation | The “DOWN\_ButtonFunction();” is programmed the same way as the UP BUTTON except it works opposite of the “UP\_ButtonFunction();” |

|  |  |
| --- | --- |
| **RIGHT AND LEFT BUTTON FUNCTION** | |
| LEFT\_ButtonFunction();  RIGHT\_ButtonFunction();  LEFT\_ButtonFunction()  {  if (LEFT\_Button.justPressed())  {  if (UI\_Navigation == 2)  {  if (DeviceParaPosition != 1)  {  DeviceParaPosition = DeviceParaPosition - 1;  }  else if (DeviceParaPosition == 1)  {  DeviceParaPosition = 1;  }  }  }  }  RIGHT\_ButtonFunction()  {  if (RIGHT\_Button.justPressed())  {  if (UI\_Navigation == 2)  {  if (DeviceParaPosition != 18)  {  DeviceParaPosition = DeviceParaPosition + 1;  }  else if (DeviceParaPosition == 18)  {  DeviceParaPosition = 18;  }  }  }  } | |
| Explanation | When user press LEFT Button, the DeviceParaPosition variable will *DECREASE* as long as the value of DeviceParaPosition != 1, if the value DeviceParaPostion == 1 then the value of it will be set to 1.  When user press RIGHT Button, the DeviceParaPosition variable will *INCREASE* as long as the value of DeviceParaPosition != 18, if the value DeviceParaPostion == 18 then the value of it will be set to 18. |

|  |  |
| --- | --- |
| **SET BUTTON FUNCTION** | |
| SET\_ButtonFunction();  SET\_ButtonFunction()  {  if (SET\_Button.justPressed())  {  if (UI\_Navigation == 0 && PROG\_Cursor\_Selection == 0 ) //PROG MENU, Operation Status Selected  {  UI\_Navigation = 1;  OperationStatus(SystemStatus);  }  else if (UI\_Navigation == 0 && PROG\_Cursor\_Selection == 1 ) //PROG MENU, Device Parameter Selected  {  UI\_Navigation = 2;  DevicePara();  }  else if (UI\_Navigation == 2)  {  UI\_Navigation = 1;  Set\_Parameter();  SystemStatus = 1;  Output\_BITS(map(RPM,0,1500,0,2047));  OperationStatus(SystemStatus);  }  }  } | |
| Explanation | The Set\_ButtonFunction(); basically is the “Enter Key” on your keyboard. This function has different if statement for different UI. If the SET Button is pressed in PROG UI, then the user can go into the selected UI.  If Set Button is pressed and the condition is (UI\_Navigation == 0 && PROG\_Cursor\_Selection == 0) then it will take the user to Operation Status UI with the parameter input of SystemStatus and set the UI\_Navigation varible to 1.  Else if the condition is (UI\_Navigation == 0 && PROG\_Cursor\_Selection == 1) then it will take the user to Device Parameter UI and set the UI\_Navigation variable to 2.  However, if the user is in the Device Parameter UI (UI\_Navigation == 2), then it will call the function Set\_Parameter(); the Set\_Parameter() function consist of a “for loop” the for loop initialize a local variable x = 0 and if x < 5 it will execute the if statement. The if statement have the condition (x<3) which sets the final parameters variable Static\_xxxx from the temporary variables that were changed. When the “for loop” is done executing its tasks, the value of RPM will be set to the MIN RPM and Initializes the system and takes the user to Operation Status UI. |

|  |  |
| --- | --- |
| **START STOP BUTTON FUNCTION** | |
| START\_ButtonFunction();  STOP\_ButtonFunction();  STOP\_ButtonFunction()  {  if (STOP\_Button.justPressed())  {  if (UI\_Navigation == 1)  {  SystemStatus = 0;  OperationStatus(SystemStatus);  Output\_BITS(0);  }  }  }  START\_ButtonFunction()  {  if (START\_Button.justPressed())  {  if (UI\_Navigation == 1)  {  SystemStatus = 1;  OperationStatus(SystemStatus);  Output\_BITS(map(RPM,0,1500,0,2047));  }  }  } | |
| Explanation | START\_ButtonFunction will set the   * SystemStatus = 1 * Calls OperationStatus(SystemStatus) function which takes SystemStatus as input * Call the Output\_BITS() function with the input of MAPPED RPM VALUE   STOP\_ButtonFunction will set the   * SystemStatus = 0 * Calls OperationStatus(SystemStatus) function which takes SystemStatus as input * Calls the Output\_BITS() function with the input RPM of 0 |