**HERIOT-WATT UNIVERSITY**

**B31DG-EMBEDDED SOFTWARE**

**ASSIGNMENT- 1.**

**NAME: GAUTAMI ALAGARSAMY**

**HWU-ID: H00423618**

**SUBMISSION DATE:17.2.2023**

**GITHUB REPOSITORY:**

**https://github.com/GautamiAlagarsamy/Embedded-Software.git**

**PROBLEM STATEMENT:**

This program generates two waveform signals, **Signal A** and **Signal B**.

The waveform consists of repeated cycles, and **signal** **B** should be 50 microseconds.

There are two switch inputs to generate the signals, **Switch1** and **Switch2**, and four parameters, **a**, **b**, **c**, and **d**.

* If **Switch1** is 0, it enables the stream of pulses.
* If **Switch1** is 1, it disables the stream of pulses.
* Similarly, if **Switch2** is 0, the program runs in normal mode,
* If **Switch2** is 1, it runs with a new waveform cycle.

The new waveform cycle parameters are defined as follows:

* **a** is the width of the first pulse, the 2nd pulse is **a + 50 microseconds**, the 3rd pulse is **a + 100 microseconds**, and so on.
* **b** is the width of the space between pulses (all spaces are the same).
* **c** is the number of pulses in a block.
* **d** is the space between pulse blocks.

**PARAMETER CALCULATION:**

* Each student will have a set of (a, b, c, d, and mode) parameters. Calculate your set of parameters using the following information.  
  The alphabet is numbered -  
  a=1, b=2, c=3, ... m=13  
  n=13, o=12, p=11, ... z=1
* Using the first 5 letters of your surname (repeat the last letter if less than 5) the parameters are  
  calculated as follows :  
  Parameter Example (James Herd)  
  a = first letter \* 100μS a = h \* 100μS = 8 \* 100μS = 800μS  
  b = second letter \* 100μS b = e \* 100μS = 5 \* 100μS = 500μS  
  c = third letter + 4 c = r + 4 = 9 + 4 = 13  
  d = fourth letter \* 500μS d = d \* 500μS = 4 \* 500μS = 2.0mS  
  mode = remainder (fifth letter/4) +1 mode = rem(d / 4) +1 = rem(4 / 4) +1 =1

**FIRST NAME: GAUTAMI**

**LAST NAME: ALAGARSAMY**

**Parameters calculation : A=1,L=12,A=1,G=7 and A=1.**

1. Parameter 'a' calculated as first letter\*100 = 1\*100 = 100=>**a = 100**
2. Parameter 'b' calculated as second letter\*100 = 12\*100 = 1200 => **b = 1200**
3. Parameter 'c' calculated as third letter+4 = 1+4=5=> **c = 5**
4. Parameter 'd' calculated as fourth letter\*500 = 7\*500 = 3500=> **d = 3500**
5. Mode is calculated as rem(fifth letter/4) + 1 = rem(1/4) + 1 = 2. Hence, we use **mode 2.**

**PYTHON CODE: ESP32 - C3 DevKit-02U**

const int ledPin1=9; // initialize the input pin of LED1

const int ledPin2=10; //initialize the input pin of LED2

const int pushButtonPin1=4; //initialize the input pin of pushbutton switch1

const int pushButtonPin2=5; //initialize the input pin of pushbutton switch2

int  pushButtonState1=0; //assign the state of pushbutton switch1

int  pushButtonState2=0; //assign the state of pushbutton switch2

//Last Name:ALAGARSAMY  First five alphabets of surname A=1,L=12,A=1,G=7, A=1

int a = 100;// parameter 'a' calculated as first letter\*100 = 1\*100=100

int inv = 4900;// parameter used in mode '2' where we reverse the signal, hence highest value is a+(c-1)\*b 100+(5-1)\*1200=4900

int b = 1200; // parameter 'b' calculated as second letter\*100 = 12\*100 = 1200

int c =5; // parameter 'c' calculated as third letter+4 = 1+4=5

int d = 3500; // parameter 'd' calculated as fourth letter\*500 = 7\*500 = 3500

// mode is calculated as rem(fifth letter/4) + 1 = rem(1/4) + 1 = 2. Hence we use mode 2

void setup()

{

  pinMode (pushButtonPin1, INPUT );

  pinMode (pushButtonPin2, INPUT );

  pinMode (ledPin1,OUTPUT);

  pinMode (ledPin2,OUTPUT);

 }

 void loop() {

 pushButtonState1=digitalRead(pushButtonPin1);

 pushButtonState2=digitalRead(pushButtonPin2);

 if(pushButtonPin1== HIGH)

//condition if switch1 is high the LEDs are in OFF condition

 {

   digitalWrite(ledPin1, LOW);

   digitalWrite(ledPin2, LOW);

   }

   else

   {

   if(pushButtonPin2 == LOW)

//condition if switch2 is low the output waveform is normal.

   {

     digitalWrite(ledPin2, HIGH);

     delay(50);

     digitalWrite(ledPin2, LOW);

     }

     if(pushButtonPin2 == HIGH)//condition if switch2 is high the new output waveform is generated.

     {

      digitalWrite(ledPin1, HIGH);

      delay(a);// parameter a=width of the first pulse

      c=c-3;

      digitalWrite(ledPin1, LOW);

      delay(b);

      a = a+50;// the 2nd pulse b is a + 50 microseconds

      digitalWrite(ledPin1, LOW);

      delay(d);

      }

      else

      {

        digitalWrite(ledPin2, HIGH);

        delay(50);

        digitalWrite(ledPin2, LOW);

        for(int i =0; i<c;i++)// c is the number of pulses in a block

        {

          digitalWrite(ledPin1, HIGH);

          digitalWrite(ledPin2, LOW);

          delay(inv);

          digitalWrite(ledPin1, LOW);

          delay(b);

          inv = inv-50;

          digitalWrite(ledPin2, LOW);

          }

          digitalWrite(ledPin1, LOW);

          delay(d); //d is the space between pulse blocks.

          d=d/2;b=b/2;

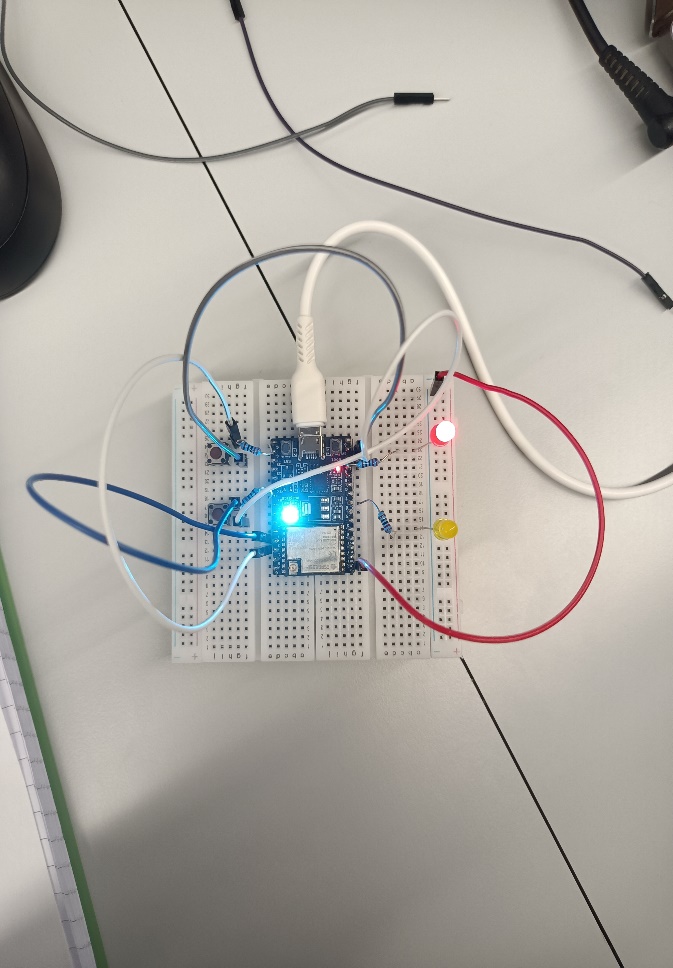
          digitalWrite(ledPin2, LOW);

          }

          }

          }

**CIRCUIT DIAGRAM &EXPERIMENT**

** A picture containing text, indoor, electronics, computer

Description automatically generated**

**Diagram

Description automatically generated with medium confidence**

**OUTPUT DISCUSSION:**

The connections are given in the breadboard with a push button switch to pin 4 and push button 2 to pin 6 of the ESP32 C3 board. The LED1 and LED2 connections are given to Pin 1 and Pin3 respectively. The supply and ground connections are given as per the circuit diagram. The Port and boards are selected, and the program is dumped into the ESP32 C3 board. The program is compiled and debugged to get the output signal.

The push buttons are pressed to generate the stream of pulses on both LEDs as per commands.

The new waveform cycle parameters are defined as follows: a is the width of the first pulse, the 2nd pulse is a + 50 microseconds, the 3rd pulse is a + 100 microseconds, and so on. b is the width of the space between pulses (all spaces are the same). c is the number of pulses in a block, and d is the space between pulse blocks.