Department of Engineering and Physical Science

Embedded Software
Assignment 1

Sylvain Jannin

H00387879

Summary

Table of figures

1. Context and electronic design

The goal of this assignment is to send signals from an ESP32. The first thing to do is to know the characteristic of our signal. The family name is "*Jannin*" so:

- a → 10 → 1 000 µs
- b → 1 → 100 µs
- c → 17
- d → 13 → 6 500 µs
- mode → 2

We have the different parameters, and we have our two mods: normal mode and mode 2.

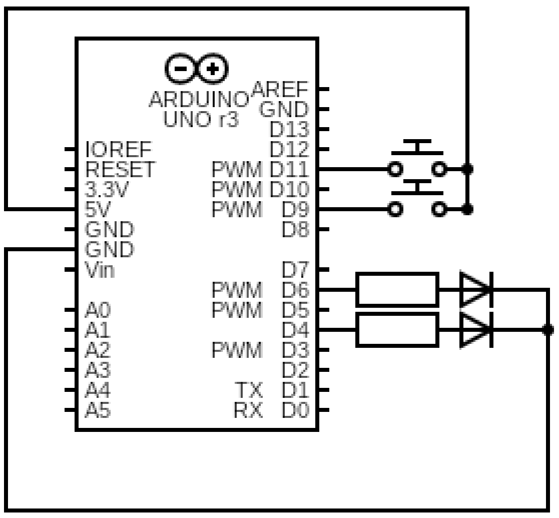


Figure 1 : electric schematic.

On the figure above there is the electric schematic, the ESP32 is represented by the Arduino Uno. There are two switches, one to enable the signal and the second one to choose the signal mode.

On the figure below, the microcontroller with the two LEDs to see the signal when the delay is in seconds and not in microseconds. There are also two switches to control the microcontroller.

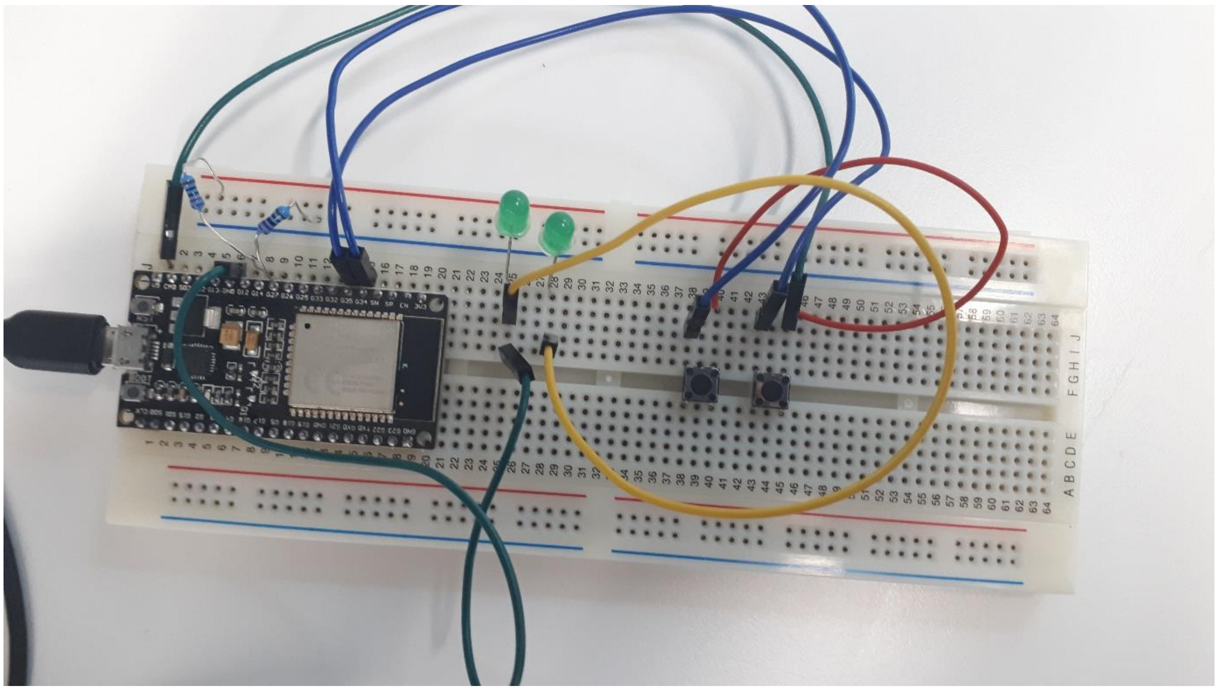


Figure 2 : electric assembly pictures.

2. Measures on Oscillocope

The below figures show the two signals in different configurations. The Yellow is signal A and the blue one is signal B. On figure 3, we can clearly see the signal B being high before signal A starts. Also, we can see the width is decreasing with the time, which is the normal because on this capture it is not the normal mode but the mode 2. On figure 4 the width is increasing with time which is the normal mode because switch 1 is pressed.

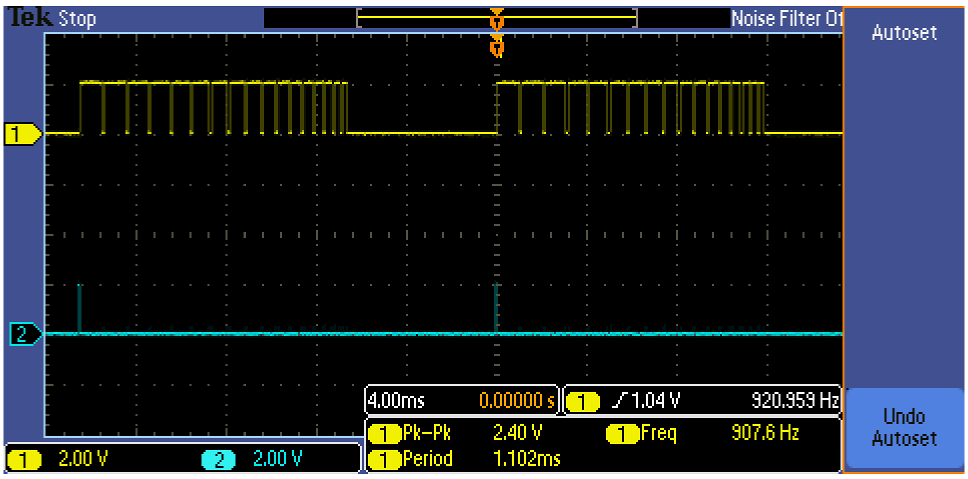
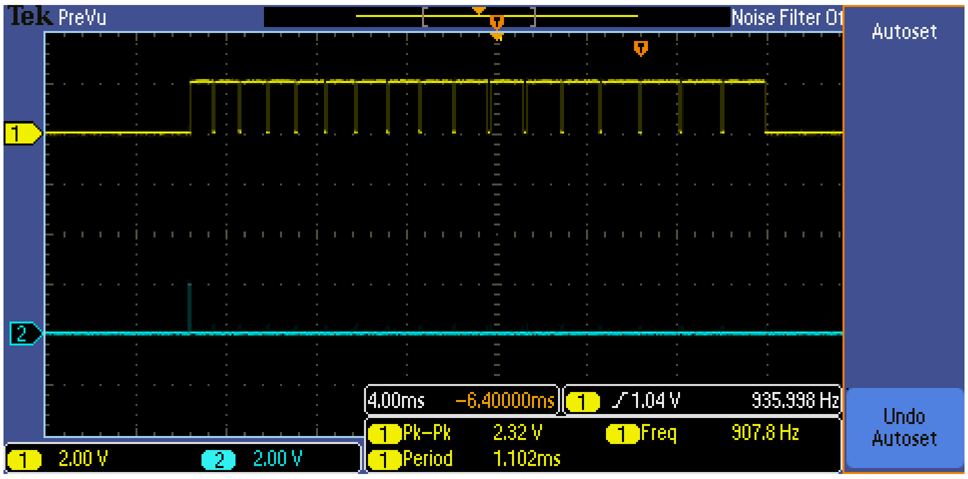


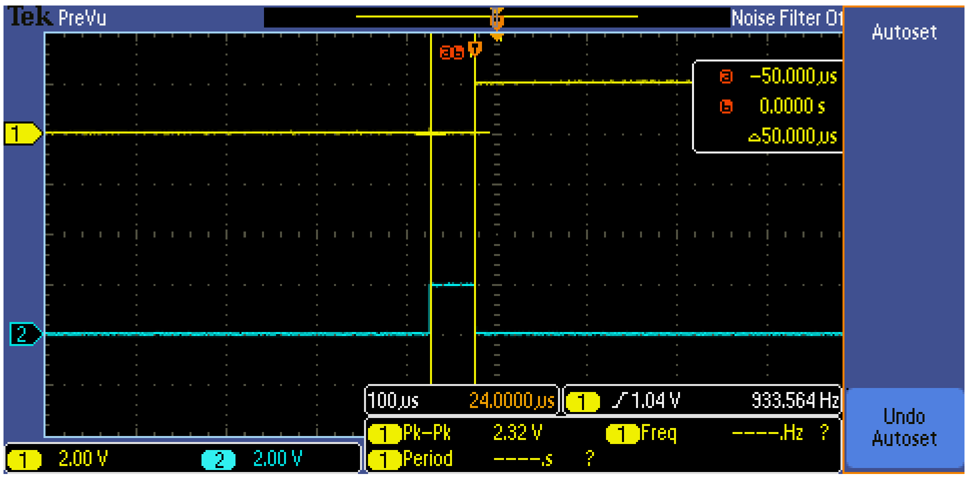
Figure 3 : Capture of both signals in mode 2 for 2 periods.



```
Tek PreVu                                Noise Filter Of
4.00ms  -6.40000ms   1  1.04 V   935.998 Hz
1 Pk-Pk   2.32 V     1 Freq      907.8 Hz
1 Period  1.102ms
1  2.00 V    2  2.00 V
```

Figure 4 :Capture of both Signal in normal mode for one 1 period.

On the Figure 5 we can check that signal B width, it is 50µs as we want.



```
Tek PreVu                                Noise Filter Of
a  -50.000µs
b  0.0000 s
Δ50.000µs
100µs  24.0000µs   1  1.04 V   933.564 Hz
1 Pk-Pk   2.32 V     1 Freq      ----.Hz  ?
1 Period  ----.s  ?
1  2.00 V    2  2.00 V
```

Figure 5 : Measure signal B width while it is high.

On the next figure, we can check the width of the first time when the signal is high. It shows the time it 990µs when we asked for 1000 µs (=a). Since during the measurement we did not zoom enough, it is not precise enough to show the signal is high for 1000µs.

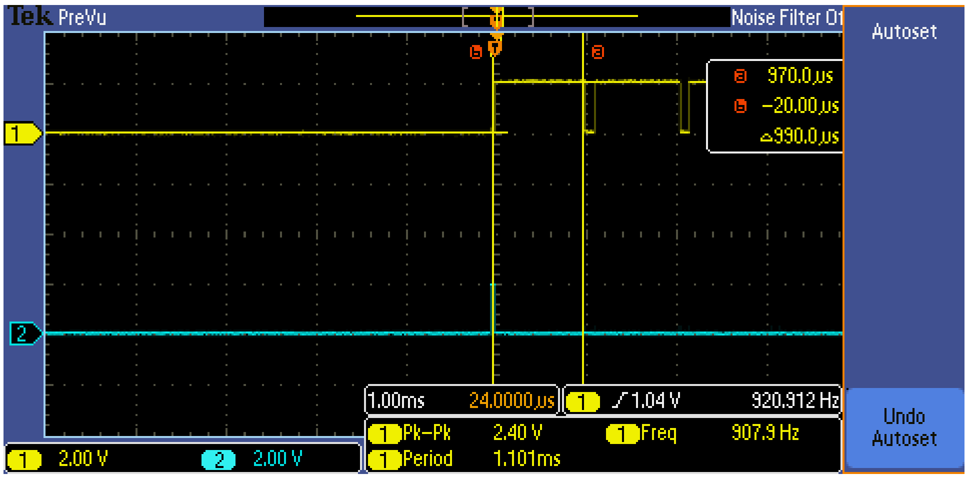


Figure 6 :measure of the first width while it is high of signal A.

Finally, when the switch 2 is pressed, there is no signal as we can see in the next figure.

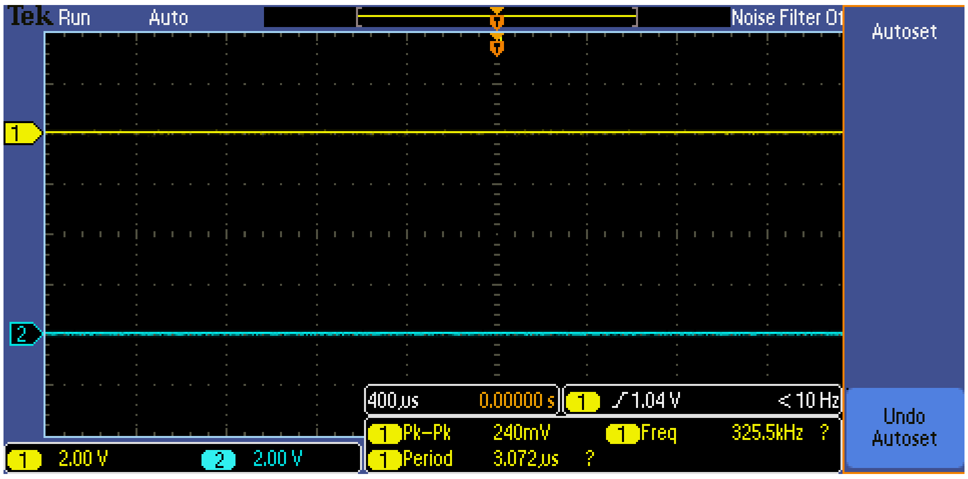


Figure 7 : No signal.

3. Program design

The diagram below shows the idea of the code. While the switch 2 is low, it first sends signal B for 50 µs then it checks the state of switch 1. If the switch 1 is low, it does the special mode when the delay between two high states is decreasing. Otherwise, the switch 1 is pressed, it does the classic mode when a is increasing. Then it does a last delay for 6 500µs (=d) before rechecking the state of switch 2.

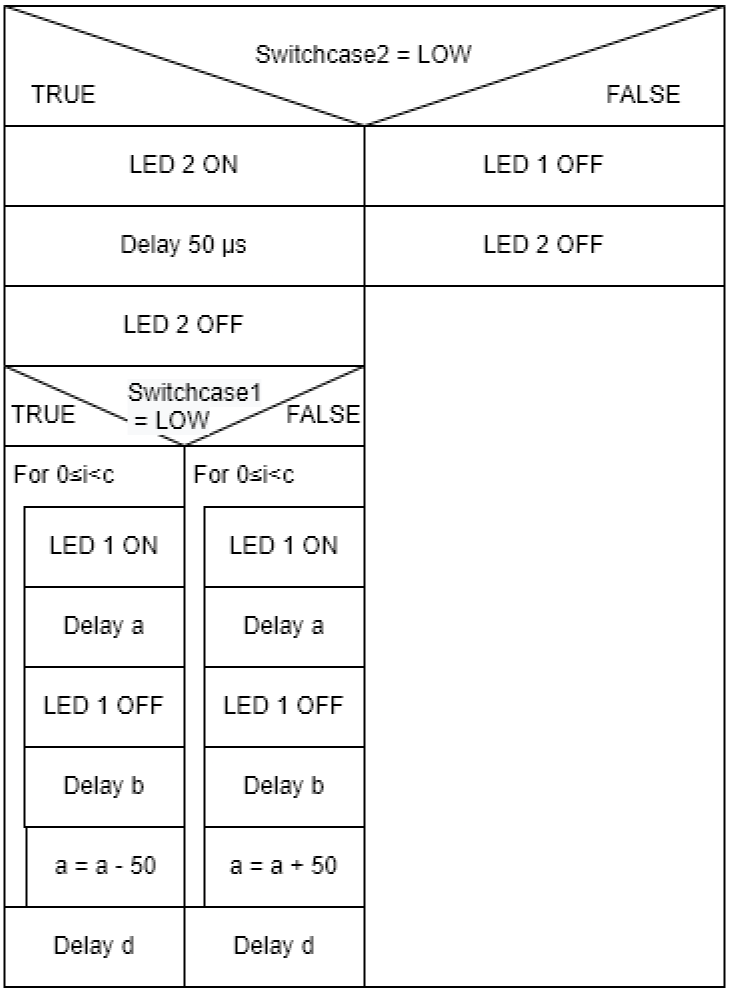
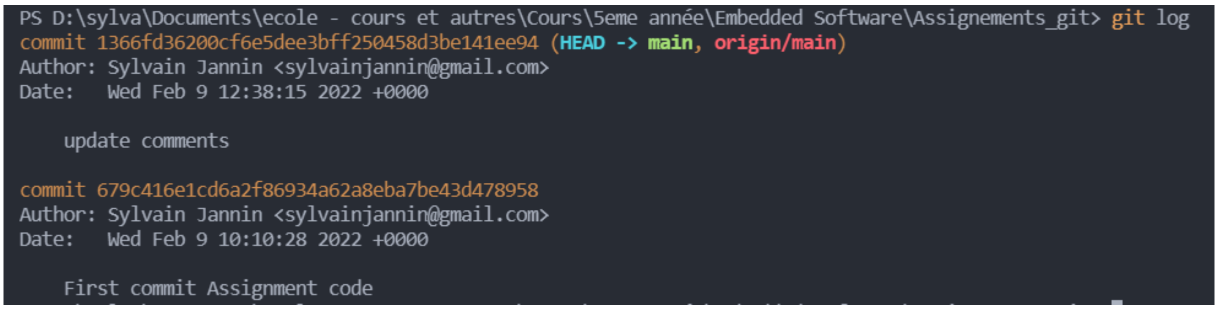


Figure 8 : Nassi-Shneiderman Diagram.

4. Git

My git repository for this class is https://github.com/SylvainJnn/EmbeddedSoftware . There is going to be only folder by assignments. On the figure below there is commit history.



```
PS D:\sylva\Documents\ecole - cours et autres\Cours\5eme année\Embedded Software\Assignements_git> git log
commit 1366fd36200cf6e5dee3bff250458d3be141ee94 (HEAD -> main, origin/main)
Author: Sylvain Jannin <sylvainjannin@gmail.com>
Date:   Wed Feb 9 12:38:15 2022 +0000

    update comments

commit 679c416e1cd6a2f86934a62a8eba7be43d478958
Author: Sylvain Jannin <sylvainjannin@gmail.com>
Date:   Wed Feb 9 10:10:28 2022 +0000

    First commit Assignment code
```

Figure 9 : Capture of git log repository.

5. Code

```
// Initialise constant, the number represents the pin where the wires and conected to the
microcrontroler. The "const" keyword is added because we do not want these values to be
changed.
const int LED1 = 14;
const int LED2 = 27;
const int Switch1 = 35;
const int Switch2 = 34;

void signal_A(int a, int b, int c, int d)   //normal mode
{

  int i;                          //init variable use for the for loop
  for(i=0; i<c; i++)  //
  {
    digitalWrite(LED1, HIGH); //turn on LED1 for a µs
    delayMicroseconds(a);

    digitalWrite(LED1, LOW);  //turn off LED1 for b µs
    delayMicroseconds(b);

    a = a + 50;            //Increment variables a
  }
  delayMicroseconds(d);                   //do a last wait
}

void signal_mod2(int a, int b, int c, int d)  //second mode
{
  int i;                          //init variable use for the for loop
  for(i=0; i<c; i++)
  {
    digitalWrite(LED1, HIGH);//turn on LED1 for a µs
    delayMicroseconds(a);
    digitalWrite(LED1, LOW);//turn off LED1 for b µs
    delayMicroseconds(b);
    a = a - 50;             //change variable
  }
  delayMicroseconds(d);
}

void signal_B()
{
  const int delay_value = 50;     //for the delay, we want a constant one
  digitalWrite(LED2, HIGH);       //send a signal for 50µs
  delayMicroseconds(delay_value); //wait 50 µS
  digitalWrite(LED2, LOW);        //stops the signal by putting it LOW
}


void setup() //set the input and output pins
```

```
{
  pinMode(Switch1, INPUT);  //set pins 34 and 35 as input
  pinMode(Switch2, INPUT);

  pinMode(LED1, OUTPUT);    //set pins 14 and 27 as output
  pinMode(LED2, OUTPUT);

}

void loop()
{
  int Switchstate1 = digitalRead(Switch1);    //create varaible to make it easier to read
  int Switchstate2 = digitalRead(Switch2);

  int a, b, c, d;
  a = 1000 ;
  b = 100 ;
  c = 17;
  d = 6500 ;

  if(Switchstate2 == LOW)                       //if the first switch is on
  {
    signal_B();                                  //send the first signal
    if(Switchstate1 == LOW)                     //check if siwtch 2 is high
    {
      signal_mod2(a, b, c, d);                   //if yes, send the second mod
    }
    else                                         //otherwise send the normal mode
    {
      signal_A(a, b, c, d);
    }
  }
  else                                           //if not, turn everything off
  {
    digitalWrite(LED1, LOW);
    digitalWrite(LED2, LOW);
  }
}
```