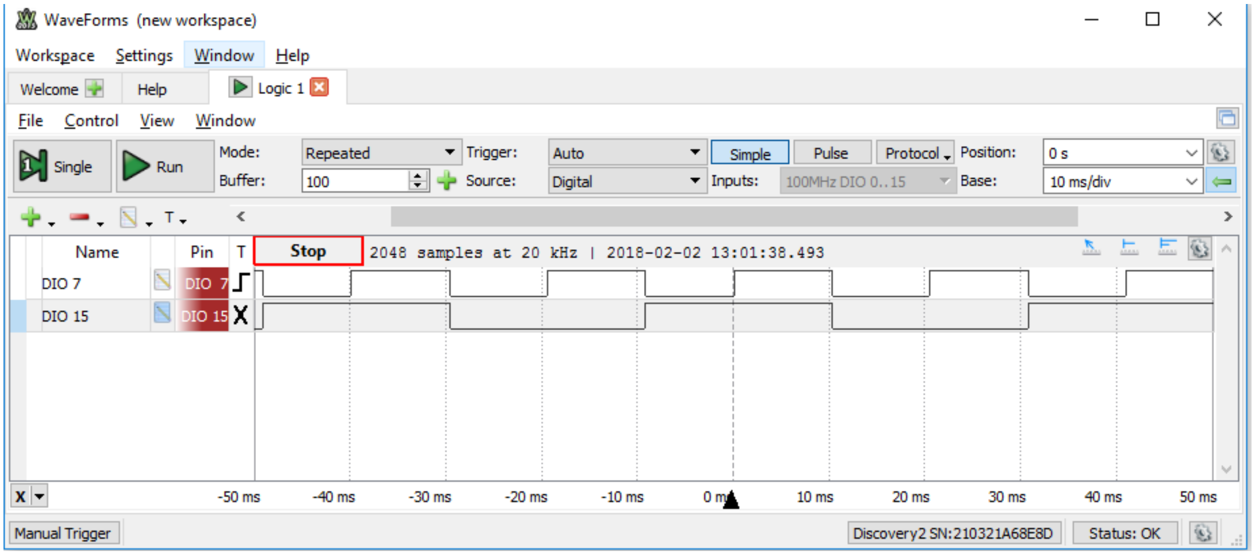
ECE 241 Lab 2 Report Spring 2018
Name: Graham Wood Lab Section A

Objective: Familiarization with the Digilent Logic Analyzer.

Prelab: The code was written and uploaded to the website. It was altered to toggle the pins at 10ms and 20ms.

Part 1) Part one consisted of uploading the additional code to the Arduino and then setting up the software and Digilent logic Analyzer. In order to set up WaveForm correctly the Position needed to be set to 0s and the base was originally set to 10ms. This is done in order to capture the 10ms bit changes our program was written to perform. The DIO 7 and 15 pin of the DLA were connected to port 12 and 13 respectively. After all was set up, the program was able to record what was going on the Arduino. As you can see by the screen shot, DIO 7's value changes every 10ms and DIO15's value changes every 20 ms.

The for part one can be found down below in the appendix.

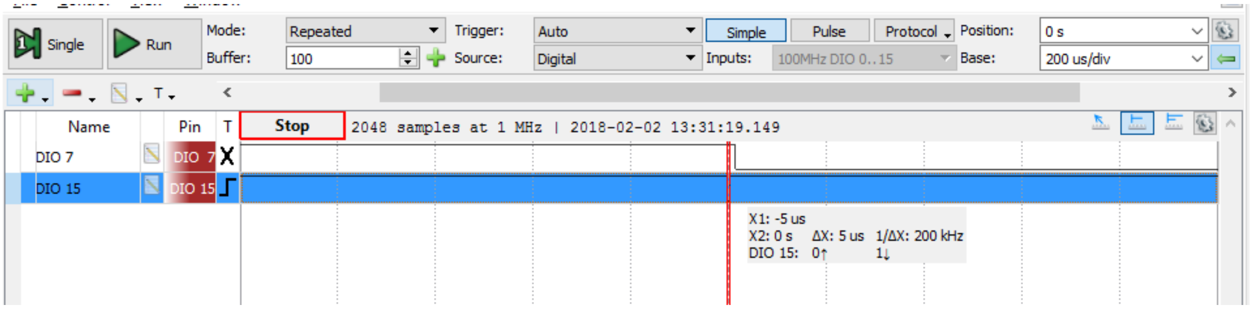

WaveForms (new workspace)
Workspace Settings Window Help
Welcome Help Logic 1
File Control View Window
Single Run
Simple Pulse Protocol
Name Pin T
DIO 7 DIO 7
DIO 15 DIO 15
-50 ms -40 ms -30 ms -20 ms -10 ms 0 ms 10 ms 20 ms 30 ms 40 ms 50 ms
Manual Trigger

Mode: Repeated
Buffer: 100
Trigger: Auto
Source: Digital
Inputs: 100MHz DIO 0..15
Position: 0 s
Base: 10 ms/div

Stop 2048 samples at 20 kHz | 2018-02-02 13:01:38.493

Discovery2 SN:210321A68E8D Status: OK

Part 2) In part 2, the program is altered so that the 10ms pin is now flipped off and then immediately back on every 20ms. Immediately meaning as fast as the digitalWrite method allows. The resulting WaveForm can be seen below. Included is the measurement of how fast the digitalWrite procedure can toggle the light. The code is also included in the appendix below.

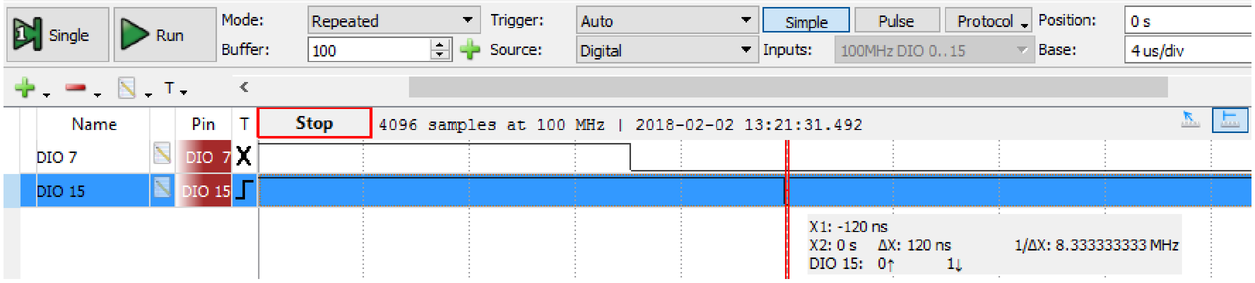


Mode: Repeated
Buffer: 100
Trigger: Auto
Source: Digital
Inputs: 100MHz DIO 0..15
Position: 0 s
Base: 200 us/div

Stop 2048 samples at 1 MHz | 2018-02-02 13:31:19.149

X1: -5 us
X2: 0 s ΔX: 5 us 1/ΔX: 200 kHz
DIO 15: 0↑ 1↓

Part 3) In part 3, the program has done away with the digitalWrite method and is now using pulsing the pin directly through PORTB. The program is then measured to see how fast a light can be pulsed using this forced method. As illustrated by the screenshot the forced pulse is much faster at toggling the light. This is because the DigitalWrite method is a built in procedure that has more overhead, therefore taking more processing power, which causes it to take longer to perform the same operation.



Mode: Repeated
Buffer: 100
Trigger: Auto
Source: Digital
Inputs: 100MHz DIO 0..15
Position: 0 s
Base: 4 us/div

```
4096 samples at 100 MHz | 2018-02-02 13:21:31.492
```

X1: -120 ns
X2: 0 s ΔX: 120 ns 1/ΔX: 8.333333333 MHz
DIO 15: 0↑ 1↓

Appendix A) Code for Part 1

```
// =>Creates two longs that will keep track of the time elapsed since the
light has been toggled
unsigned long LedTimer;
unsigned long LedTimer3;
// =>set pin 12 and 13 to output pin and sets the LedTimers equivalent to the
amount of ms ince the program started
void setup(){
  pinMode(13, OUTPUT);
  pinMode( 12, OUTPUT);
  LedTimer = millis();
  LedTimer3 = millis();
}

//=>This method will loop as long as the program is running
void loop()
{
    //=> will check if it has been 10ms since the light has been toggled, if
so it toggles it
  if (millis() - LedTimer >= 10) {
    if (digitalRead(13) == HIGH) {
      digitalWrite(13, LOW);
    }
    else {
      digitalWrite(13,HIGH);
    }
    LedTimer3 += 10;
  }
  //checks if it has been 20ms since LedTimer3 has been toggled and if so,
toggles it
  if (millis() - LedTimer3 >= 20) {
    if (digitalRead(13) == HIGH) {
      digitalWrite(13, LOW);
    }
    else {
      digitalWrite(13,HIGH);
    }
    LedTimer3 += 20;
  }
}
```

Appendix B for part 2)

```
// =>Creates two longs that will keep track of the time elapsed since the
light has been toggled
unsigned long LedTimer;
unsigned long LedTimer3;
// =>set pin 12 and 13 to output pin and sets the LedTimers equivalent to the
amount of ms ince the program started
void setup(){
  pinMode(13, OUTPUT);
  pinMode( 12, OUTPUT);
  LedTimer = millis();
  LedTimer3 = millis();
}
//=>This method will loop as long as the program is running
void loop()
{
    //=> will check if it has been 20m since the light has been toggled, if
so it switches it to lower, and immediately back to high
  if (millis() - LedTimer >= 20) {


    digitalWrite(12, LOW);
    digitalWrite(12,HIGH);

    LedTimer += 20;
  }
  //checks if it has been 20ms since LedTimer3 has been toggled and if so,
toggles it
  if (millis() - LedTimer3 >= 20) {
    if (digitalRead(13) == HIGH) {
      digitalWrite(13, LOW);
    }
    else {
      digitalWrite(13,HIGH);
    }
    LedTimer3 += 20;
  }


}
```

Appendix C for Part 3

```
/// =>Creates two longs that will keep track of the time elapsed since the
light has been toggled
unsigned long LedTimer;
unsigned long LedTimer3;
// =>set pin 12 and 13 to output pin and sets the LedTimers equivalent to the
amount of ms ince the program started

void setup(){
  pinMode(13, OUTPUT);
  pinMode( 12, OUTPUT);
  LedTimer = millis();
  LedTimer3 = millis();
}
//=>This method will loop as long as the program is running
void loop()
```

```
{
    //=> if millis - LedTimer >= 20 then the program will for the port low
and then force it high
  if (millis() - LedTimer >= 20) {

    PORTB &= ~0x10; // Force bit 4, Pin 12, low
    PORTB |= 0x10; // Force bit 4, pin 12, high



    LedTimer += 20;
  }
  //checks if it has been 20ms since LedTimer3 has been toggled and if so,
toggles it
  if (millis() - LedTimer3 >= 20) {
    if (digitalRead(13) == HIGH) {
      digitalWrite(13, LOW);
    }
    else {
      digitalWrite(13,HIGH);
    }
    LedTimer3 += 20;
  }


}
```