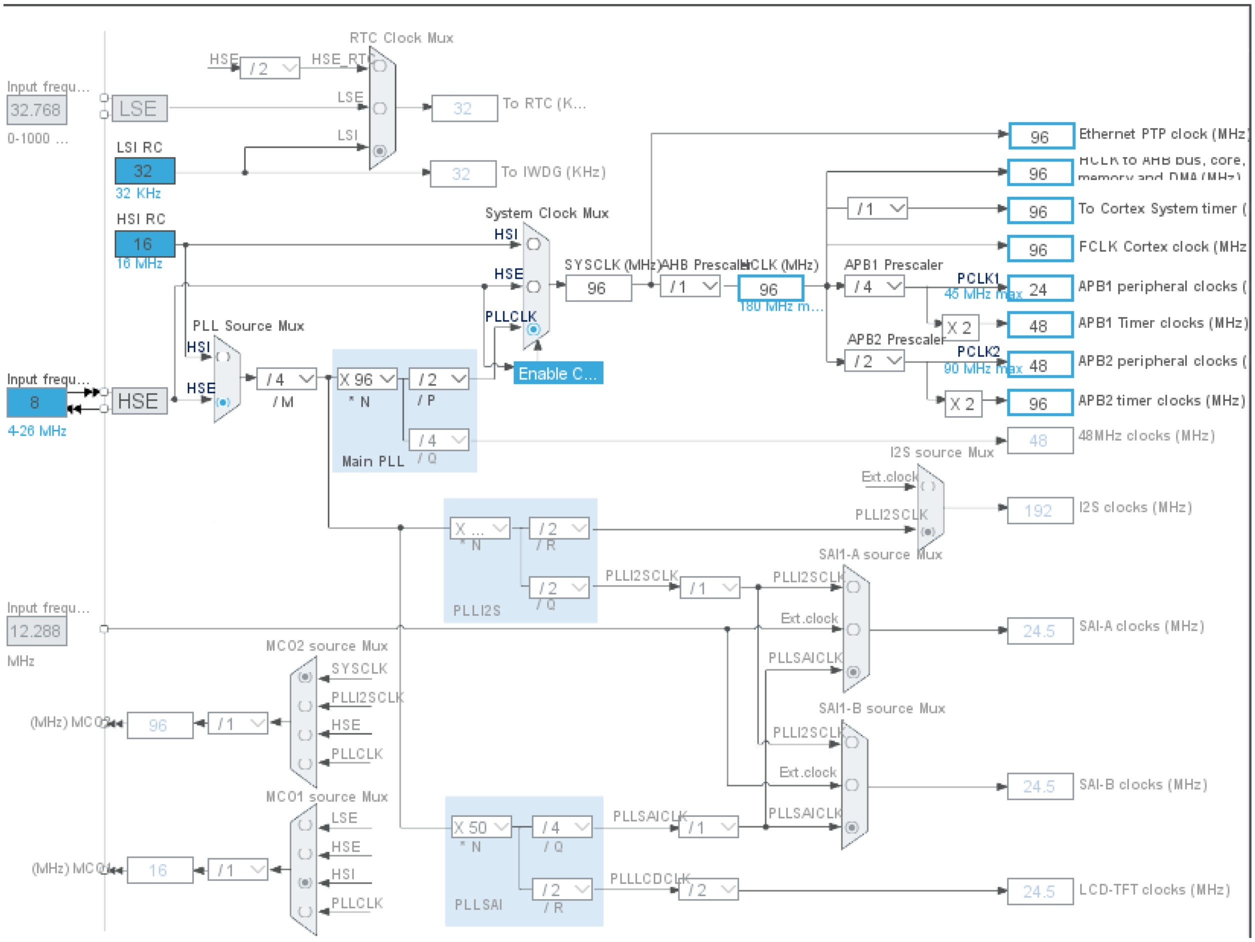
Mohit Patel

ECE 4510

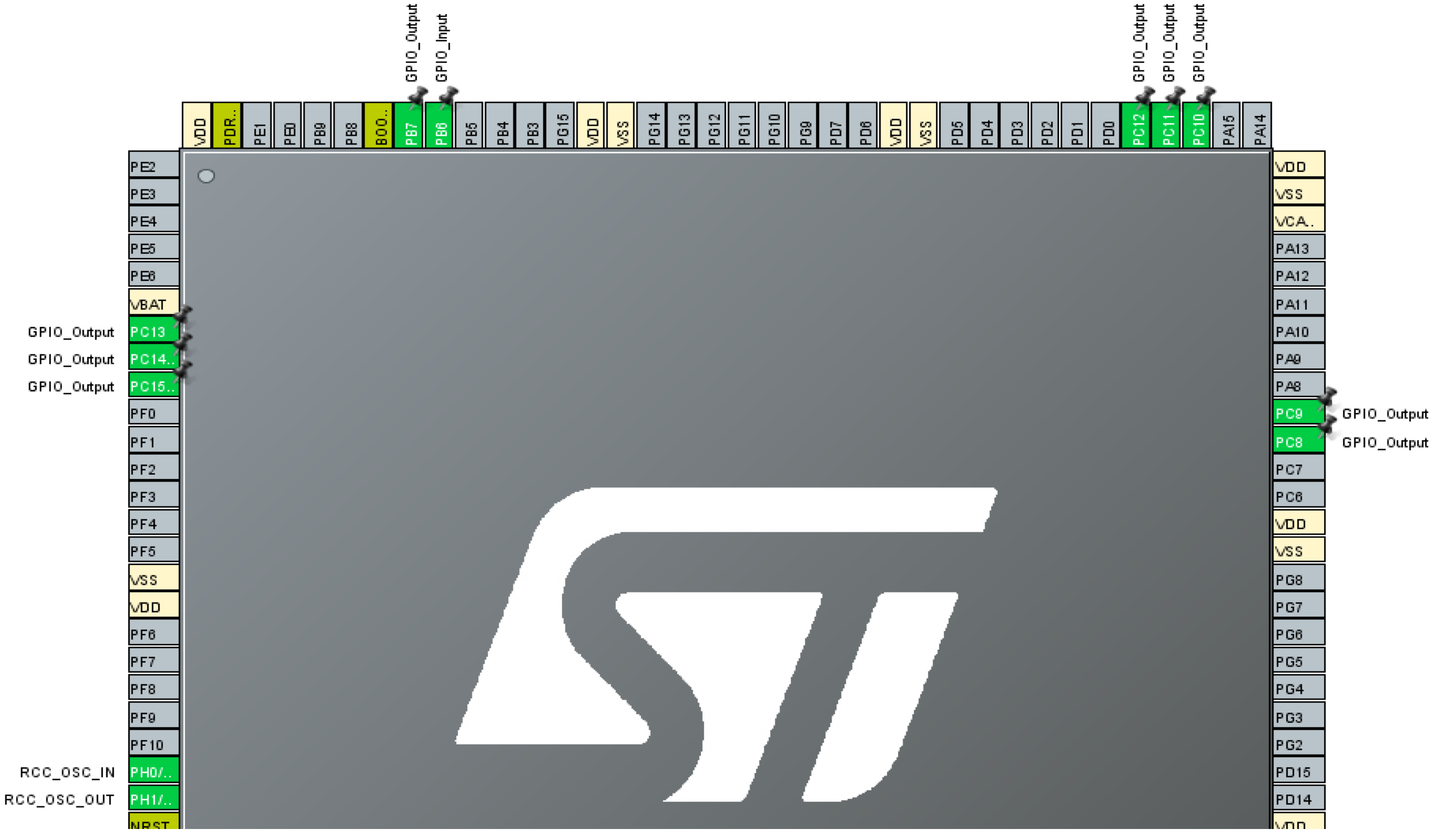
01/23/20

HW1

1.) Given the timing diagram, one possible approach that I could come up with was to use if statements and while loops for checking the start signal. But before we go more into details, I will quickly go over the clock setting and the GPIO initializations.

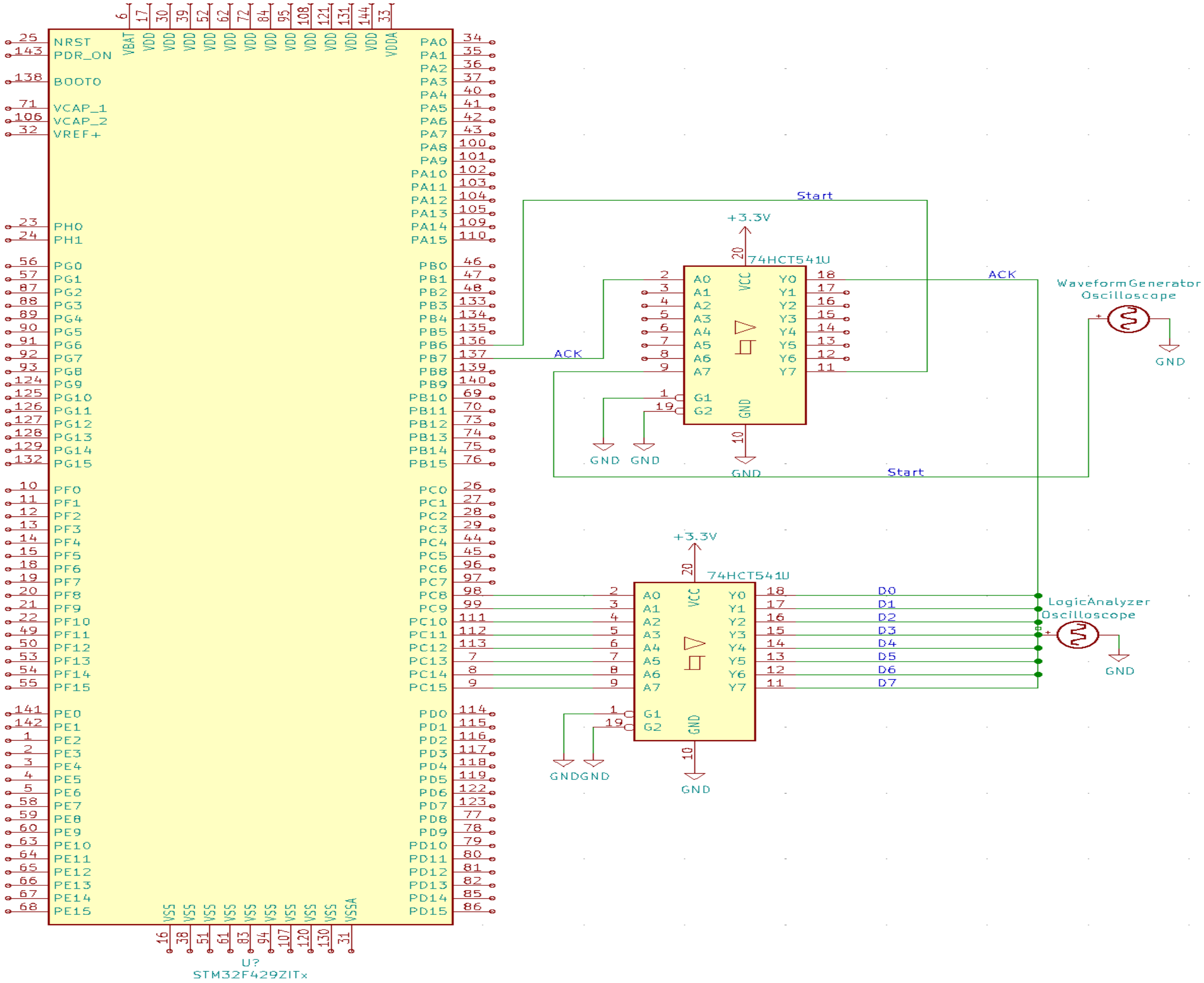


As seen in this picture the HSE frequency is 8MHz, HCLK frequency is 96 MHz and PCLK1 frequency is MHz as specified in the assignment.



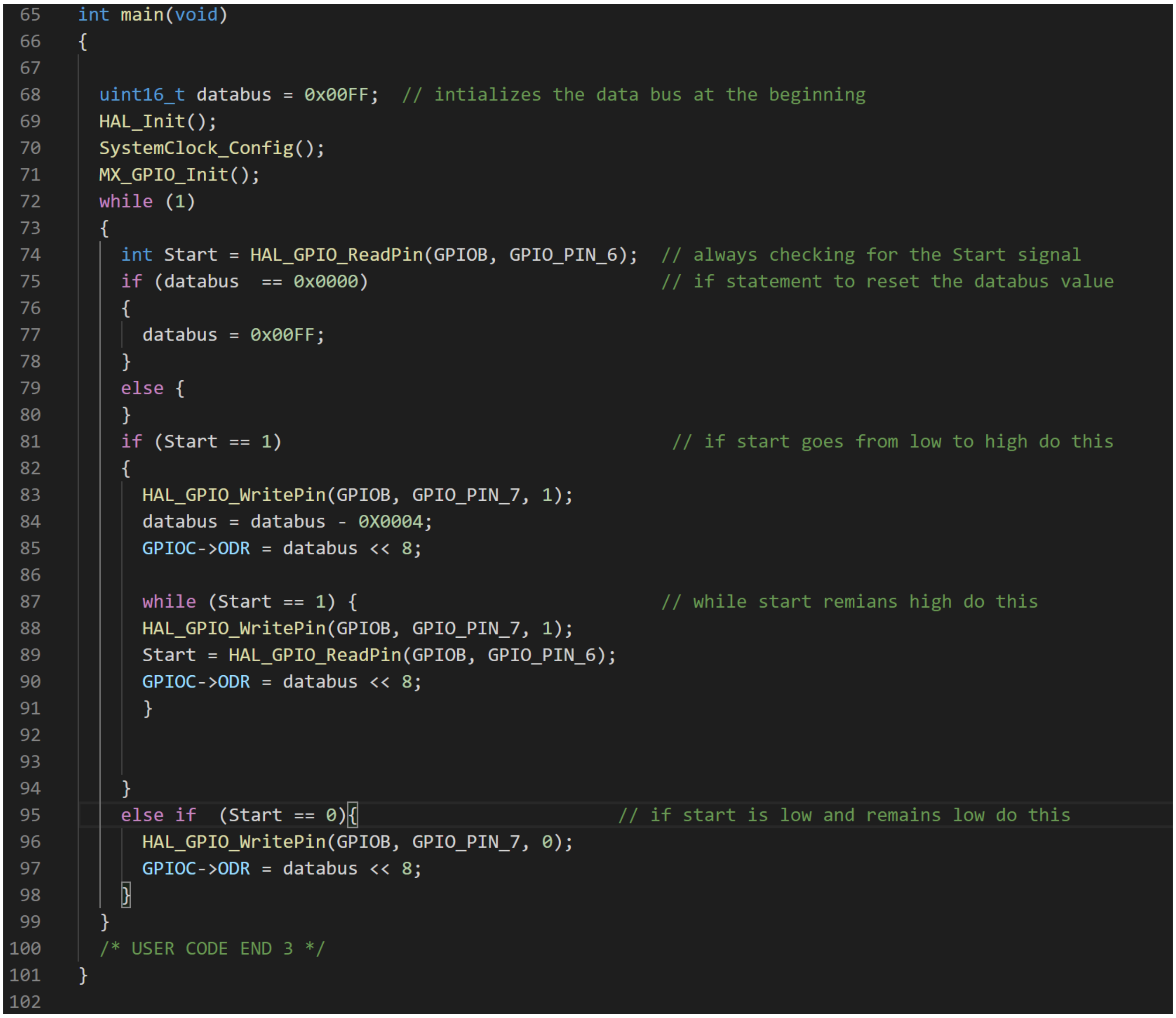
This Picture is taken from STMCUBEmx and shows the port initialization for this assignment. PC8 to PC15 are all output pins for the data bus. PB7 is the output port for the ACK signal, and PB6 is the input port for the Start signal.

2.) Schematic



The Schematic above shows the output port PC8 thru PC15 go to a non-inverting buffer before being sent into the logic analyzer. The start signal is generated from the wave gen that is inside the logic analyzer, this signal goes to a non-inverting buffer before being sent to the STM32F429ZI. The ACK signal is coming out of the board and being sent to the logic analyzer via a non-inverting buffer.

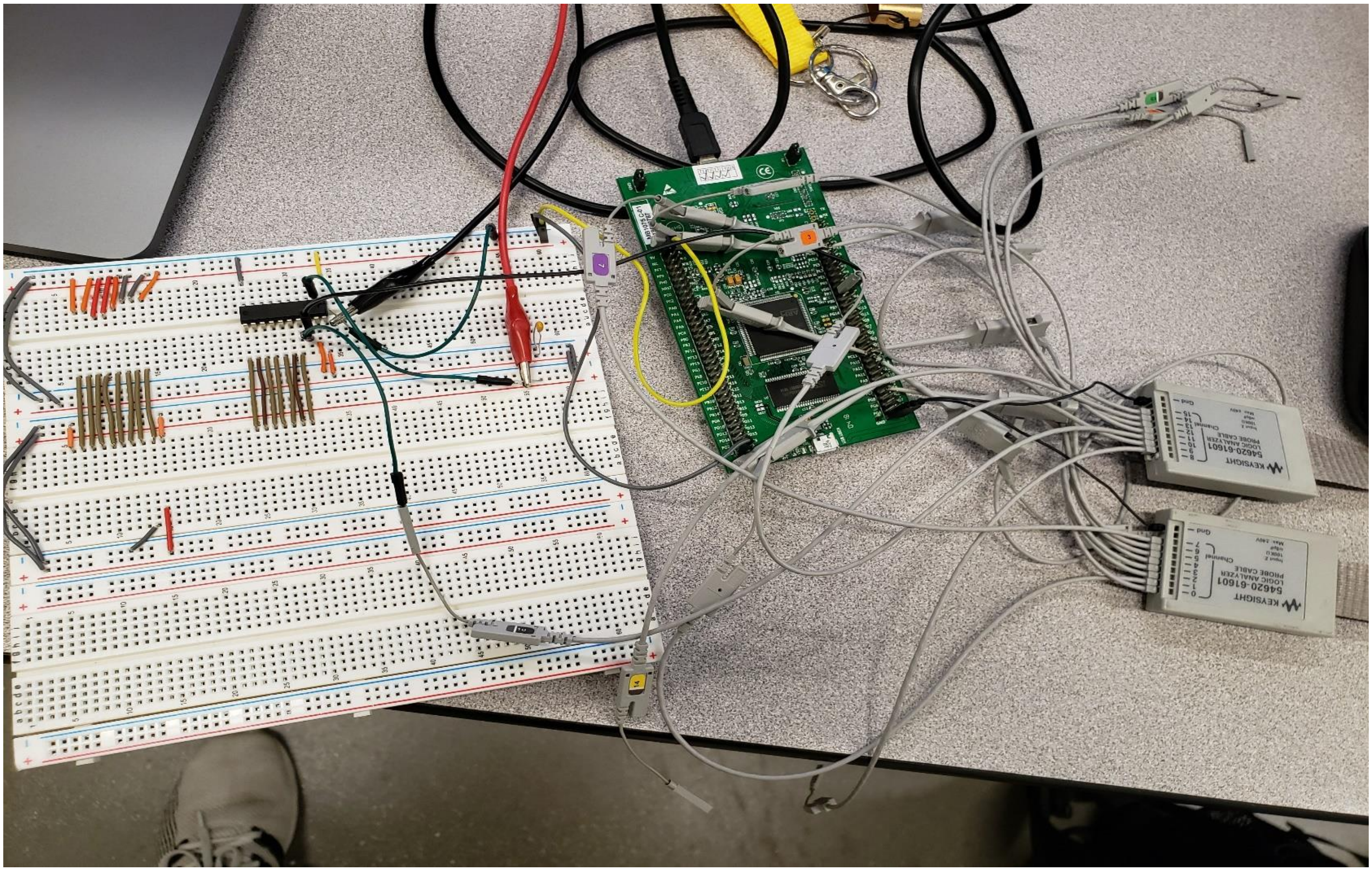
3.) CODE



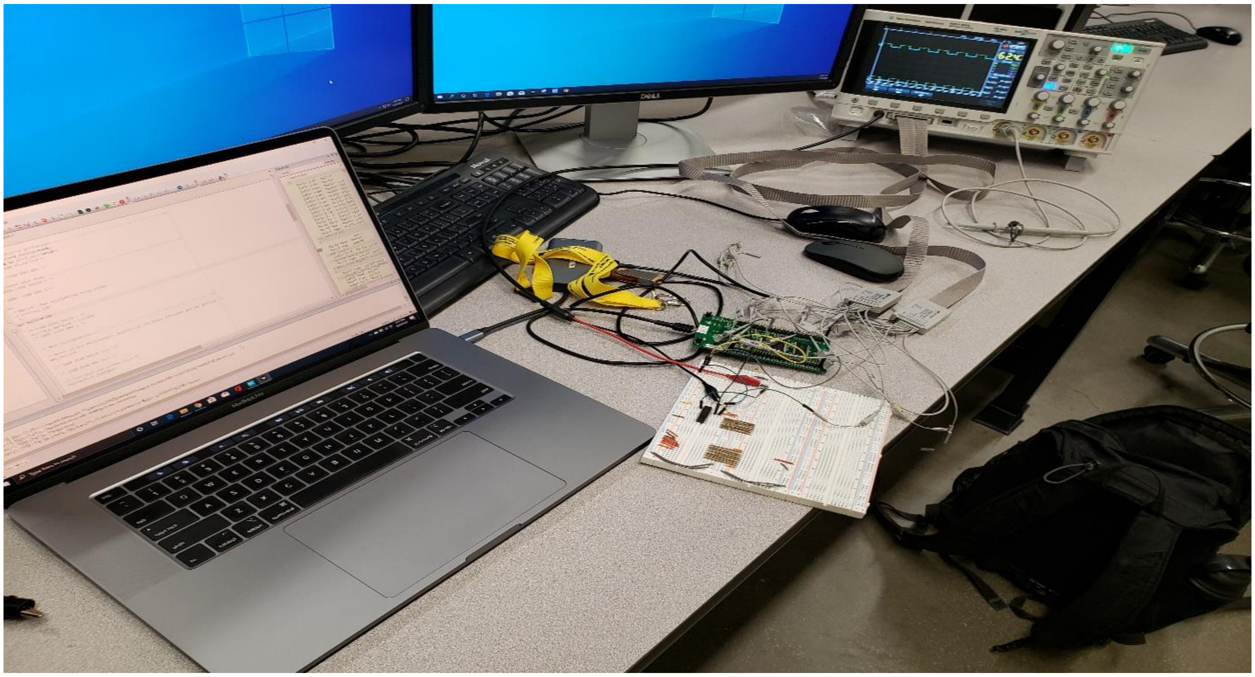
```
65  int main(void)
66  {
67
68    uint16_t databus = 0x00FF;  // intializes the data bus at the beginning
69    HAL_Init();
70    SystemClock_Config();
71    MX_GPIO_Init();
72    while (1)
73    {
74      int Start = HAL_GPIO_ReadPin(GPIOB, GPIO_PIN_6);  // always checking for the Start signal
75      if (databus  == 0x0000)                           // if statement to reset the databus value
76      {
77        databus = 0x00FF;
78      }
79      else {
80      }
81      if (Start == 1)                                    // if start goes from low to high do this
82      {
83        HAL_GPIO_WritePin(GPIOB, GPIO_PIN_7, 1);
84        databus = databus - 0X0004;
85        GPIOC->ODR = databus << 8;
86
87        while (Start == 1) {                            // while start remians high do this
88        HAL_GPIO_WritePin(GPIOB, GPIO_PIN_7, 1);
89        Start = HAL_GPIO_ReadPin(GPIOB, GPIO_PIN_6);
90        GPIOC->ODR = databus << 8;
91        }
92
93
94      }
95      else if  (Start == 0){                        // if start is low and remains low do this
96        HAL_GPIO_WritePin(GPIOB, GPIO_PIN_7, 0);
97        GPIOC->ODR = databus << 8;
98      }
99    }
100   /* USER CODE END 3 */
101 }
102
```

Line 68 initializes the DATA bus to decimal value 255. In the while (1) loop we first check for the start signal status. And then if statements are used to reset the data bus values back to 255. The nested if statement checks for the status of the input signal, and if it's high, it will make ACK signal, which is mapped to PB7, high. It will also decrement the Data bus by 4 and then sent it out to the output ports PC8 thru PC15. The while loop inside will continue to check is the start signal remains high, if this is the case, then it ill continues to have ACK high, and display the current DATA values. The else if case checks for the start signal to be low, if this is tru then ACK will be low and DATA bus will just remain unchanged.

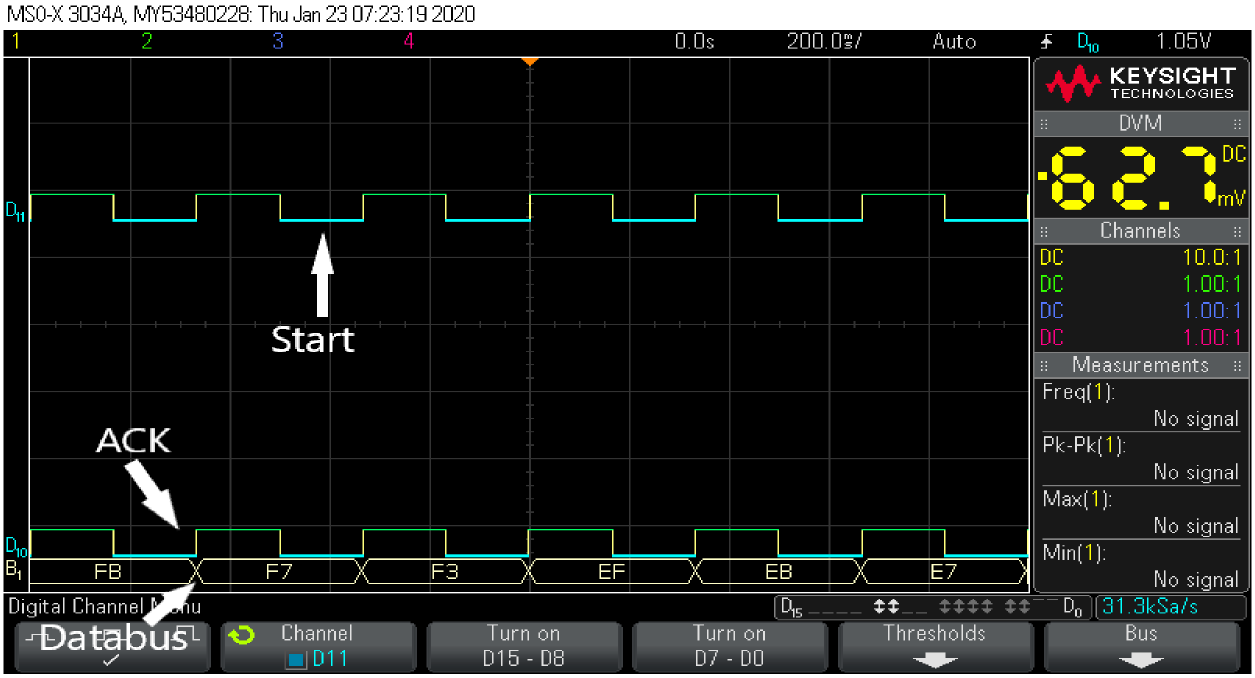
BONUS



This Picture shows the circuit setup for this assignment. Unfortunately, there was only one buffer chip, so this one buffer chip was used to take the start signal from the wave gen and send it to the board. All the output ports are directly connected to the Oscilloscope.



This picture shows the complete setup, it is easy to see how the output ports are dirclty connected to the oscilloscope from the board.



MSO-X 3034A, MY53480228: Thu Jan 23 07:23:19 2020

This is the screenshot of the oscilloscope display. Logic analyzer is used to view the data bus, ACK signal and the start signal, as labeled. As seen start and ACK are directly related to each other and that databus is decrement by 4 on the rising EDGE of the Start signal.