Binghamton University

EECE 387: Junior Design Project

Team 8

EE Miraz Sakif
CoE James Plummer
CoE Kevin William
CoE Qiang Fu

Abstract:

In this project, our team designed a battery-operated metal detector utilizing electromagnetic coupling to differentiate between three different size metal washers: small, medium, and large. Our device features an inductor consisting of two coils for enhanced detection, generating a square waveform input signal processed by an FPGA board by an Analog-to-Digital Converter. By using Vivado and Vitis, we developed software to establish thresholds for washer size differentiation and implemented a strength meter for visual indication on the LEDS. Additionally, use displayed the count of the washers being detected and their respective sizes on the Seven Segment Display. By analyzing oscilloscope waveforms, we optimized detection thresholds and managed power usage effectively. The system was verified to meet specifications, including safety considerations, and successfully demonstrated reliable detecting the lab demo.

Table of contents

Project Overview

For this project, our team has designed a battery-operated metal detector utilizing electromagnetic coupling to detect and differentiate between three different metal washers. To enhance our detector's detection capabilities, we constructed a solenoid specifically tailored for this application. The detector operates by generating a square wave signal, which serves as the input to our FPGA board, likely facilitated through an Analog-to-Digital Converter (ADC). With the use of Vivado and Vitis, we developed software to run on our FPGA board. The software was programmed to establish three distinct thresholds, allowing us to differentiate between the sizes of the washers being detected. Additionally, we implemented a visual indicator on the FPGA board: a strength meter. This strength meter illuminates based on the size of the washer detected and its proximity to the solenoid metal detector. As a metal object approaches the detector, the amplitude of the square wave signal decreases, reaching zero when a larger metal washer makes direct contact with the coil. Through this design and software implementation, our metal detector provides both accurate detection and visual feedback to the user.

Requirements

- The project should be battery operated, e.g. six rechargeable batteries (7.2V), six regular batteries (9V), or a single 9V cell. We can extract 7.2V-9V using standard electronic components, such as resistors, Zener diodes, etc.
- The design must ensure electromagnetic response in the presence of metal, coil(s) must be implemented with magnetic wire and experimentally characterized to find their resistance and inductance values, as was previously done in the lab experiments.
- The design must use the BASYS board as the main controller.
- When each type of a metal object is detected, it must be indicated by lighting up specific symbol(s) on one of the displays; this should be shown as long as the object is detected. The total number of objects detected should be displayed, as well as the number of each type. A strength meter should be implemented to indicate the strength of the detection signal for the current object; the strength meter should indicate over the full range of the detection signal.
- The design must use electronic circuits/microcontroller/FPGA to generate any waveforms needed for your design, such as pulses or sinusoidal voltages.
- The design must use standard electronic components.
- The design may need to use datasheets for certain parts, such as transistors, Zener diodes, etc., to find parameters needed for analysis.
- All parts of the system must be designed for safety; even under failure conditions. As a safety precaution, the system may not use lithium-based battery technologies (e.g. Li-ion/Li- Pol); while these batteries offer high energy densities, they can pose a fire hazard if incorrectly designed or used.
- The design must be demonstrated in the lab.
- After the design is complete, perform electrical stress analysis of the components.

Block Diagram

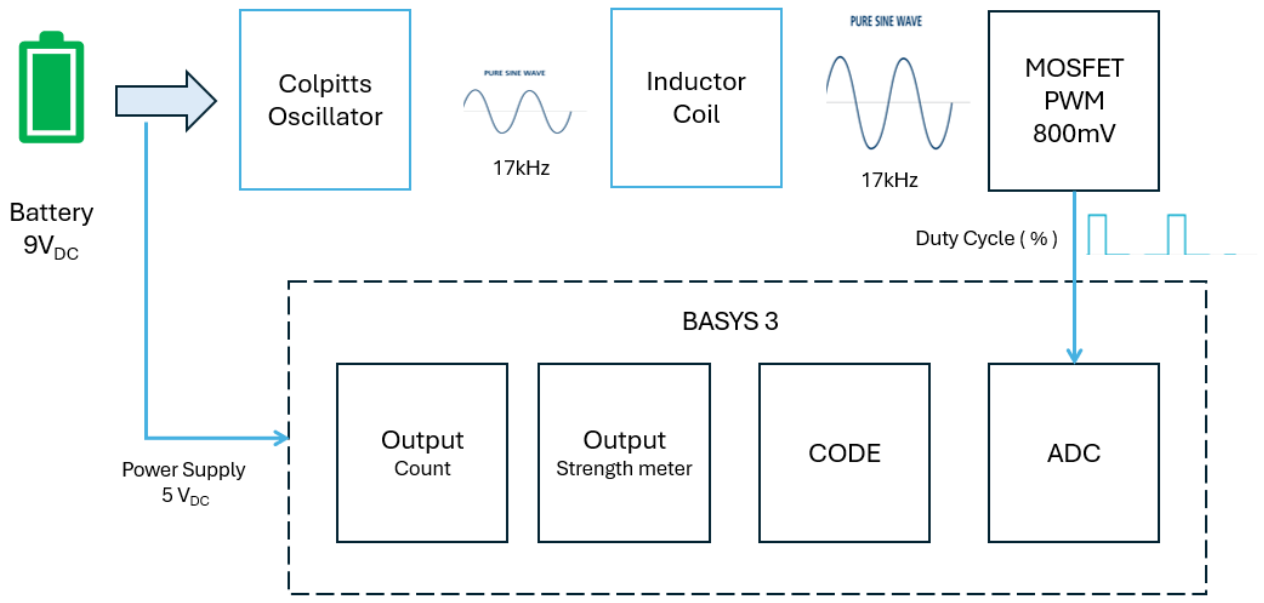


Figure 1: Project Block Diagram

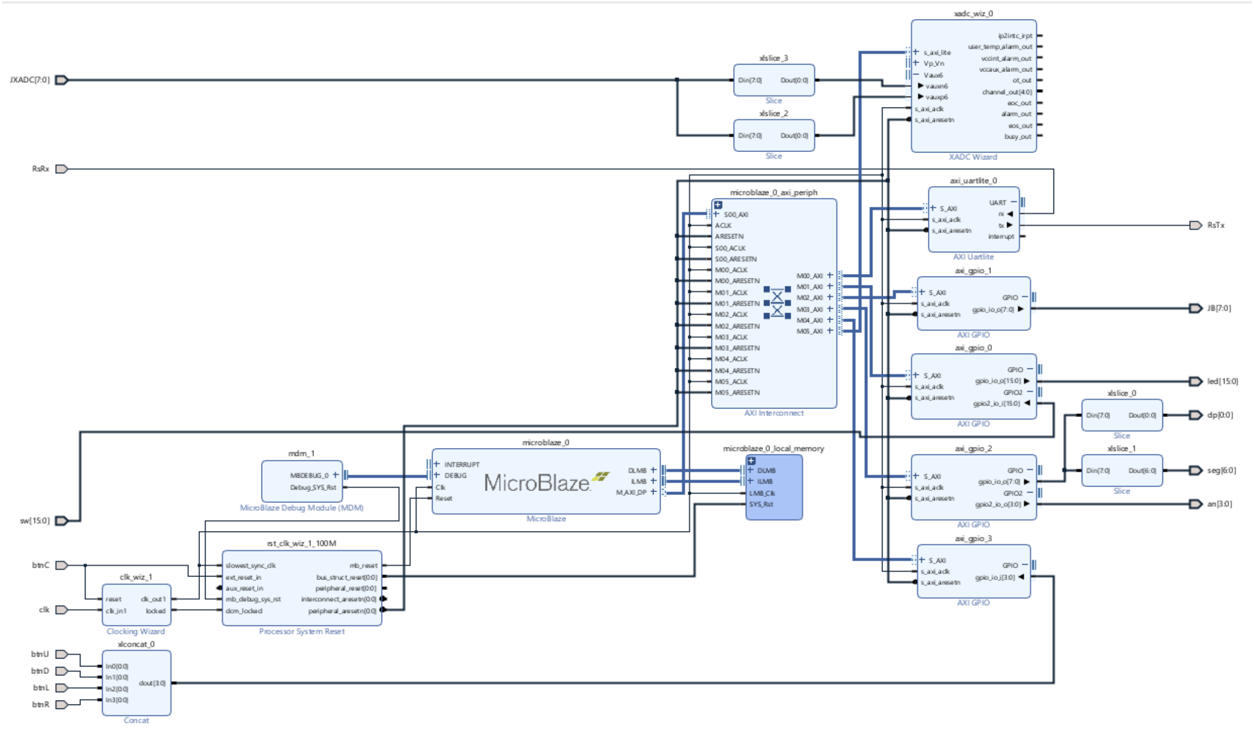


Figure 2: Vivado Block Diagram: Analog to Digital Converter

Technical Design Description

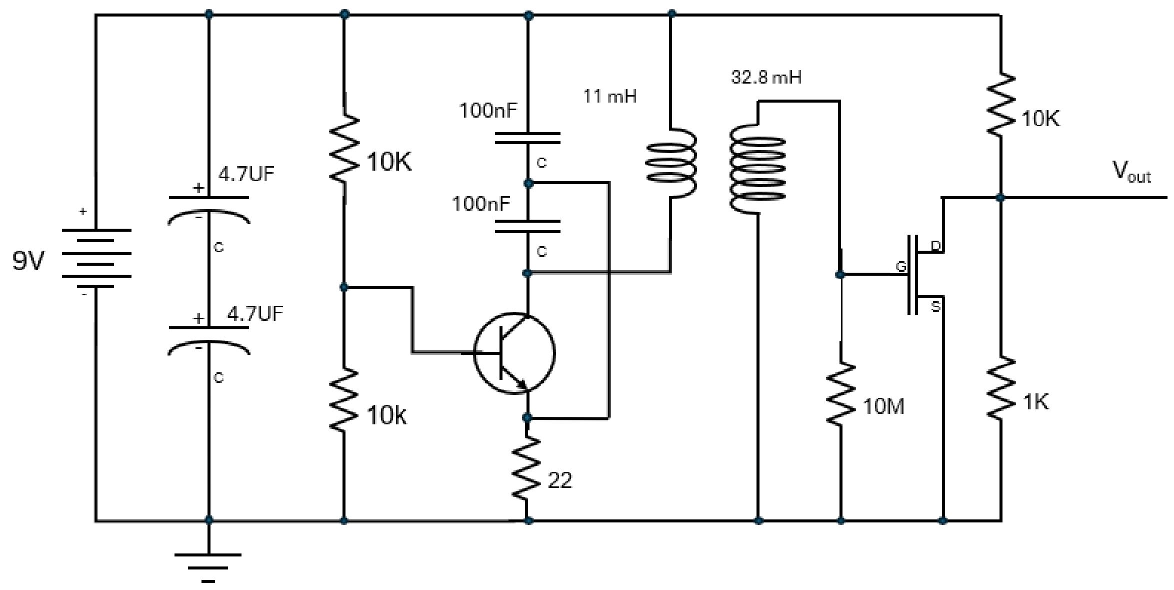


Figure 3: Circuit Diagram

Oscillator Circuit:

We implemented Colpitts oscillator, which is a type of LC oscillator used in electronic circuits to generate sinusoidal oscillations in our project. It uses an inductor and two series capacitors in parallel to the inductor to form a resonant tank circuit.

The output signal from the tank circuit is fed back into the input of a transistor, where it amplifies and fed back into the tank circuit. The feedback signal provides the necessary phase shift for sustained oscillation.

We implemented a high-quality factor (Q-factor) in the LC tank circuit, enabling stable and efficient oscillation at the desired frequency. Proper impedance matching between the gain device and the LC circuit is crucial to maintain oscillation without excessive damping. This ensures that the oscillator can sustain oscillations at its resonant frequency.

The resonant frequency generated by the LC circuit in a Colpitts oscillator can be determined by the series combination of the two capacitors in parallel to the inductor L. This resonant frequency ƒ can be calculated using the formula:

Where:

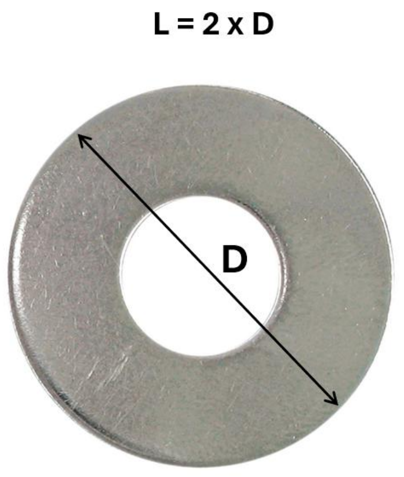
L is the inductance of the inductor.

C is the equivalent capacitance of the two capacitors in series, which is given by

This resonant frequency sets the frequency of oscillation for the Colpitts oscillator. By adjusting the values of the capacitors and the inductor, we can tune the oscillator to operate at different frequencies within its range.

Inductor Coil:

We designed an inductor to detect the metal in our project. We made as responsively as possible to detect the small ferromagnetic metal washer close to it. We made the coil with a diameter close to the diameter of medium size metal washers. We made the length of the coil two times larger than the diameter of the coil to create a large magnetic field around the coil.



Medium Washer

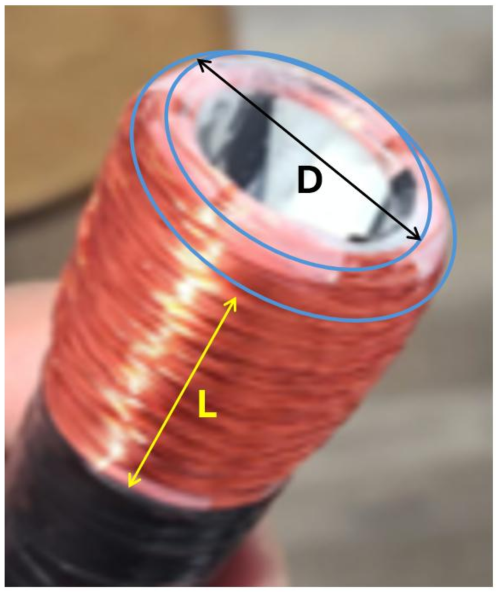


Figure 3: Inductor Coil and Washer

Design Integration

Metal detection:

Amplitude of output sine wave depends on the values inductance L, equivalent series capacitance C, emitter resistance and the biasing current. When we bring metal washer close to the air core inductor, its inductance L increases. Inductance L value is inversely proportional to the amplitude of the output signal. When L inductance increases, the amplitude of the output signal Vout decreases.

Vout = Asin wt

Different sizes of metal washer give significantly different values of L and we can see the difference in amplitude of the signal.

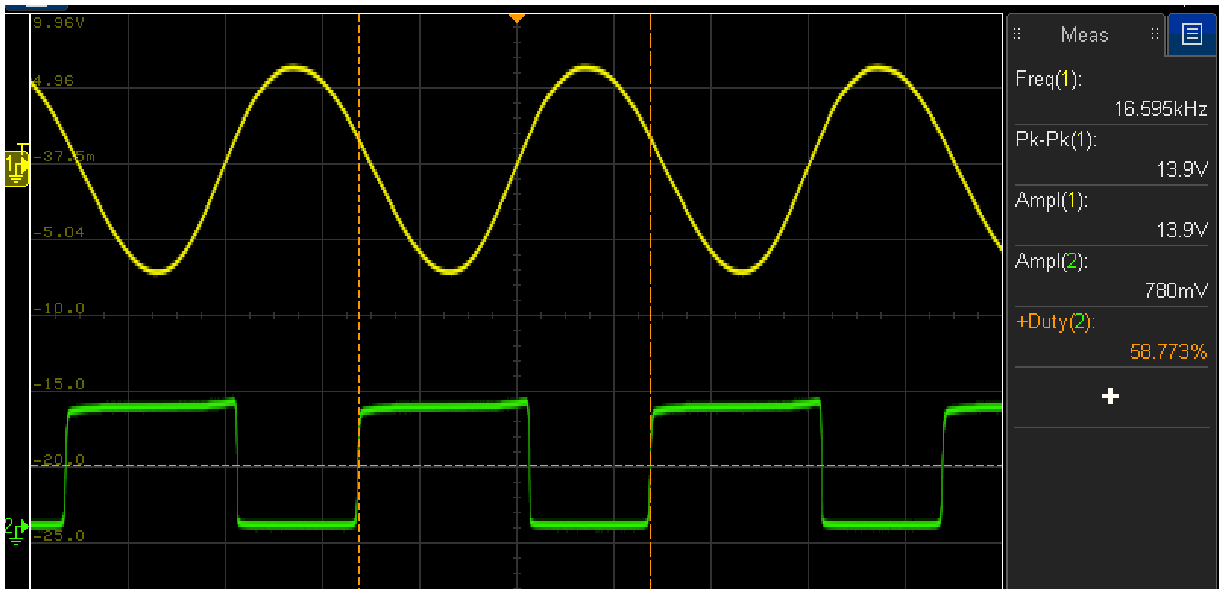


Figure 4: Waveform generated with no washer

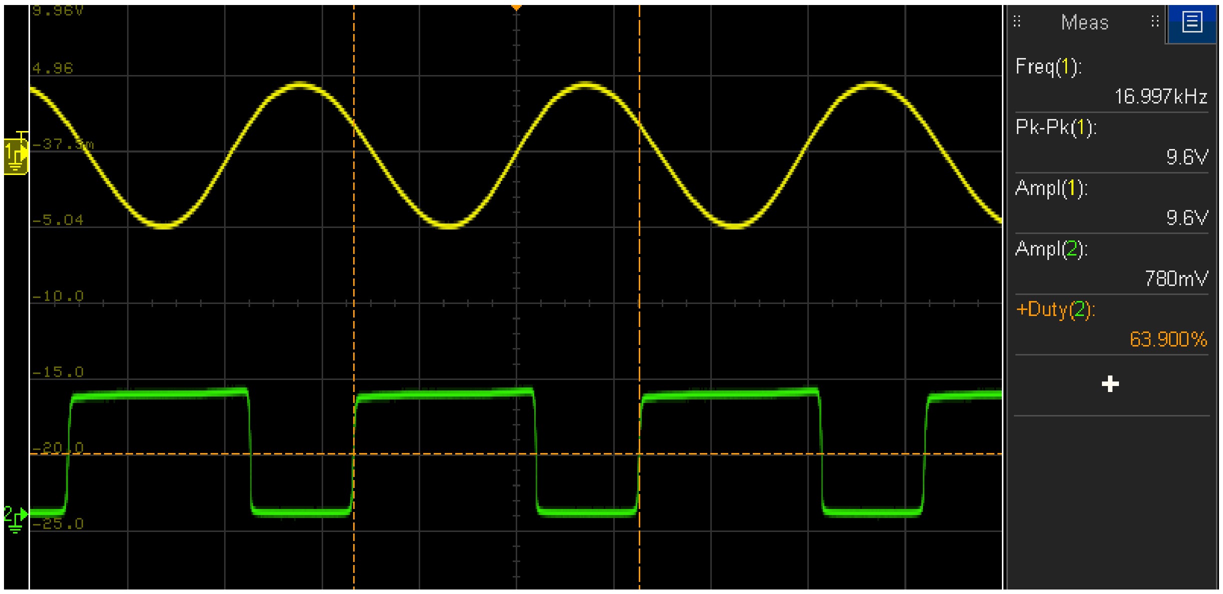
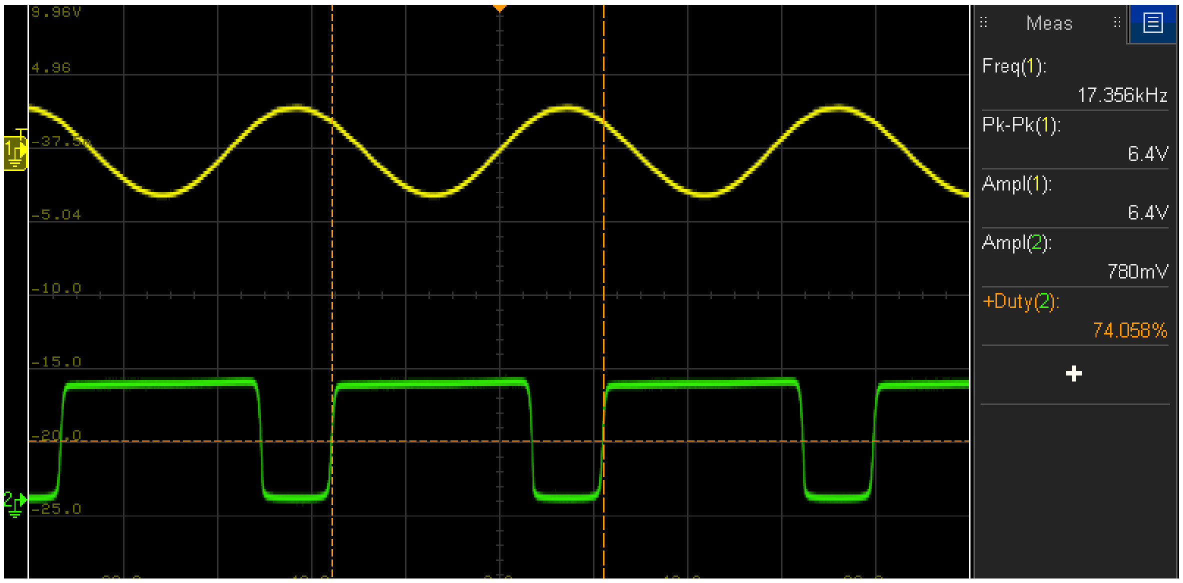
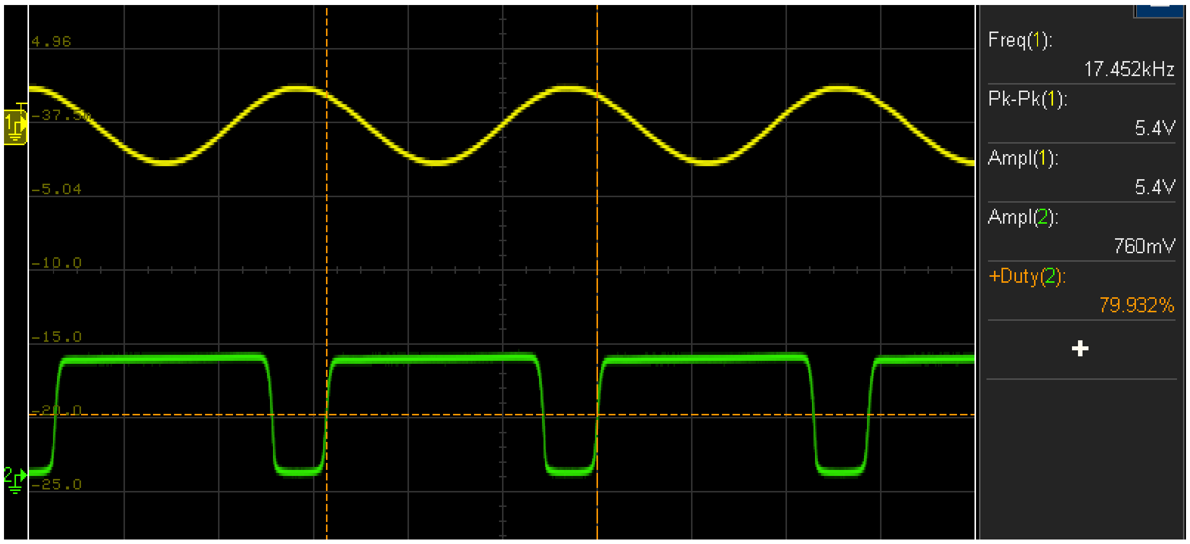


Figure 1: Waveform generated with small washer



Freq(1):
17.356kHz
Pk-Pk(1):
6.4V
Ampl(1):
6.4V
Ampl(2):
780mV
+Duty(2):
74.058%

Figure 2: Waveform generated with medium washer



Freq(1):
17.452kHz
Pk-Pk(1):
5.4V
Ampl(1):
5.4V
Ampl(2):
760mV
+Duty(2):
79.932%

Figure 3: Waveform generated with large washer

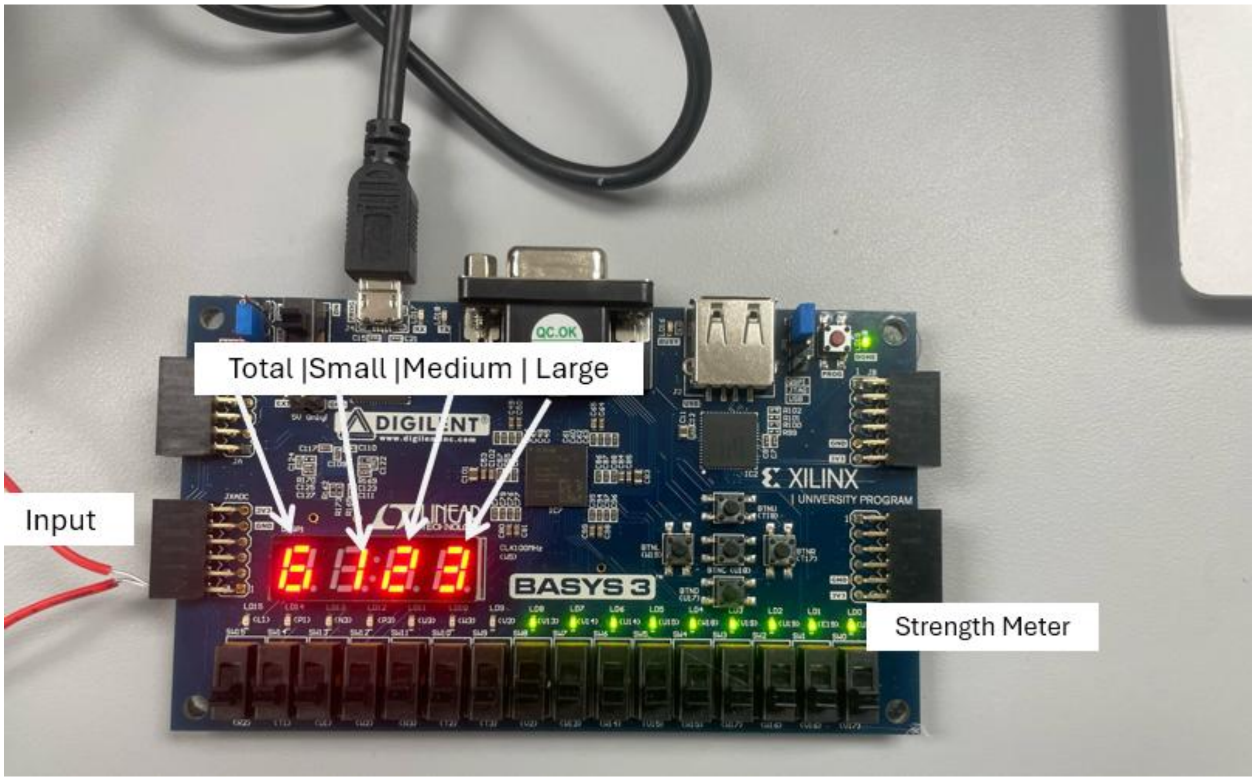


Figure 8: Basys board with code implemented

For our metal detector project, we analyzed the waveforms captured by the oscilloscope to understand the circuit's response to the three different sized washers. Initially, when no metal was detected, the waveform displayed a consistent sine wave pattern with a stable amplitude. When introducing metal to the detector, a notable decrease in amplitude and an increase in duty cycle were observed. These changes were observed using three distinct washer sizes, allowing us to establish threshold values for each. We tested these thresholds, ensuring they were as large as possible to account for variations without overlapping. During extended use of the battery power source, we observed a correlation between the battery's operational duration and increased fluctuations in the duty cycle values. To mitigate this, we disconnected the power source when not actively testing, preserving battery life, and maintaining the consistency of our readings. This approach provided reliable detection and differentiation of the washers as they were detected.

Verification

[Describe the methodology used to verify each requirement. Include supporting documentation, simulations, data sheet information, calculations, etc.]
On the EE side, he tested by doing this

Our project design meets the following requirements:

- ***Be battery operated.***
  We used a 9V battery to power the solenoid and circuit part of our design and we used 3 AAA batteries to create the required 5V to power the Basys board.

- **To ensure electromagnetic response in the presence of metal, coil(s) must be implemented with magnetic wire and experimentally characterized to find their resistance and inductance values, as was previously done in the lab experiments.**
  Inductor must be~~We~~ implemented with magnetic wire and experimentally characterized to find their resistance and inductance values, as was previously done in the lab experiments.
- **The design must use the BASYS board as the main controller.**
  We programmed our Basys board with the code for the project from Vivado and Vitis. We took the output of our circuit and used the Basys board to determine the size of the washer being detected.
- **When each type of a metal object is detected, it must be indicated by lighting up specific symbol(s) on one of the displays; this should be shown as long as the object is detected. The total number of objects detected should be displayed, as well as the number of each type. A strength meter should be implemented to indicate the strength of the detection signal for the current object; the strength meter should indicate over the full range of the detection signal.**

  We used our seven-segment display to indicate when a new metal has been detected. The left-most anode we used to count the total number of metals detected. The anode to the right of that was used to count the small washers, moving to the right again we counted the medium washers, and the large washers we counted using the right-most anode. We implemented a strength meter using the LEDs on the Basys board. When nothing is detected, no LEDs turn on. The larger the washer is being placed on the detector will cause more and more LEDs to turn on.

- **SAKIF The design must use electronic circuits/microcontroller/FPGA to generate any waveforms needed for your design, such as pulses or sinusoidal voltages.**

- **Sakif The design must use standard electronic components.**
- **Sakif The design may need to use datasheets for certain parts, such as transistors, Zener diodes, etc., to find parameters needed for analysis.**

- **All parts of the system must be designed for safety; even under failure conditions. As a safety precaution, the system may not use lithium-based**

battery technologies (e.g. Li-ion/Li- Pol); while these batteries offer high energy densities, they can pose a fire hazard if incorrectly designed or used.
Our system used a safe power source; 3x Triple A Alkaline Batteries to produce 5 volts, which was the voltage our Basys Board could operate safely.

- **The design must be demonstrated in the lab.**
  The project was presented in Monster Lab. The project correctly detected and identified 14/15 of the various metal washer tests; the small washer was detected 5 times, the medium washer was detected 4 times, and the large washer was detected 6 times.

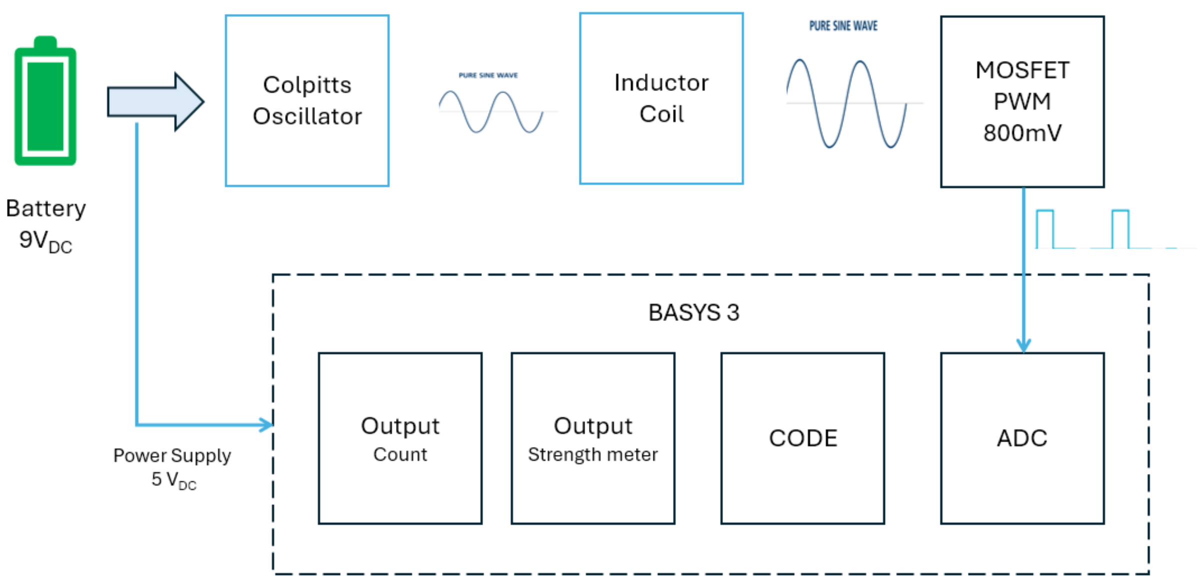
- **Sakif After your design is complete, you need to perform electrical stress analysis of your components.**

Power Budget Analysis

[Provide a critical review of how well the design met the specifications. Analyze any deviations and discuss potential improvements.]

Sakif

Software/Hardware Description



The output of our circuit went into the JXADC port on our Basys3 board. This port is connected to the XADC wizard in our Vivado code. This would take the waveform coming in and convert it into a 16-bit number. This number would represent the relative voltage in the port at the time of sampling. To find the duty cycle from this number we would add 500 samples together and find the average. The higher the average the higher the duty cycle would be. Using this concept, we found three thresholds by printing the 16-bit number from JXADC on our computer and testing with the three washers. The problem with the battery was found here, because of the variation in duty cycle, the values we are using are also changing requiring use to adjust our thresholds respectfully.

Appendices

Bill of Materials

[Provide a comprehensive bill of materials listing all the components used in the project.]
Sakif Please add your stuff here

Basys 3 Board

Source Code

```
#include <stdio.h>
#include "xil_printf.h"

#define DELAY_UNIT 81
#define LEDS  (*(unsigned volatile *)0x40000000)
#define SW    (*(unsigned volatile *)0x40000008)
#define JB    (*(unsigned volatile *)0x40010000)
#define DPSEG (*(unsigned volatile *)0x40020000)
#define AN    (*(unsigned volatile *)0x40020008)
#define BTN   (*(unsigned volatile *)0x40030000)
#define JXADC1_CH1 (*(unsigned volatile *)0x44a00258)

void seg_disp(uint8_t data[4]) {
  const uint8_t disp_lut[16] = {
    0b00111111, // 0
    0b00000110, // 1
    0b01011011, // 2
    0b01001111, // 3
    0b01100110, // 4
    0b01101101, // 5
    0b01111101, // 6
```

```
        0b00000111, // 7
        0b01111111, // 8
        0b01101111, // 9
        0b01110111, //a
        0b01111100, //b
        0b00111001, //c
        0b01011110, //d
        0b01111001, //e
        0b01110001, //f
    };

    static uint8_t digit = 0;

    // Display the current digit on the 7-segment display
    DPSEG = ~disp_lut[data[3 - digit]];

    // Select the current digit by enabling its corresponding anode
    AN = ~(1 << digit);

    // Move to the next digit
    digit = (digit + 1) % 4;
}

void delay_ms(unsigned t)
{
    unsigned cntr1, cntr2;
    while(t--)
    {
        for(cntr1=0; cntr1<100; cntr1++){
            for(cntr2=0; cntr2<DELAY_UNIT; cntr2++){}
        }
    }

}

_Bool up_button_press() {
    static _Bool state = 0;
    _Bool flag = 0;
    if (!state && (BTN & (1 << 0)))
        flag = 1;
    state = BTN & (1 << 0);
    return flag;
}

_Bool down_button_press() {
    static _Bool state = 0;
    _Bool flag = 0;
    if (!state && (BTN & (1 << 1)))
        flag = 1;
    state = BTN & (1 << 1);
    return flag;
}

_Bool left_button_press() {
    static _Bool state = 0;
    _Bool flag = 0;
    if (!state && (BTN & (1 << 2)))
        flag = 1;
    state = BTN & (1 << 2);
    return flag;
}

_Bool right_button_press() {
    static _Bool state = 0;
    _Bool flag = 0;
    if (!state && (BTN & (1 << 3)))
        flag = 1;
    state = BTN & (1 << 3);
    return flag;
}

_Bool metal_detected(uint16_t state_cnt)
{
    static _Bool state = 0;
    _Bool flag = 0;
    if(!state && state_cnt==1000)
        flag = 1;
    state = state_cnt==1000;
    return flag;

}
int main()
{
    unsigned nothing_threshold = 31000;
    unsigned small_threshold = 32000;
    unsigned med_threshold = 35000;
    unsigned large_threshold = 40000;
    uint16_t state_cnt_small = 0;
    uint16_t state_cnt_med = 0;
    uint16_t state_cnt_large = 0;
```

```
uint16_t strength;
uint16_t strength_cnt=0;
unsigned strength_tot;
uint16_t strength_avg;
uint8_t total_cnt = 0;
uint8_t small_cnt = 0;
uint8_t med_cnt = 0;
uint8_t large_cnt = 0;
uint8_t data[4];
unsigned val = 1;
while (val){

  delay_ms(1);
  strength = (JXADC1_CH1>>0);
  if(strength_cnt<=500)
  {
      strength_cnt++;
      strength_tot+=strength;
      if(strength_cnt == 500)
      {
        strength_avg = strength_tot/500;
        strength_tot=0;
        strength_cnt=0;

      }
  }
  //Turn off Strength meter LEDS
  LEDS&=~(1>>0);
  LEDS&=~(1<<1);
  LEDS&=~(1<<2);
  LEDS&=~(1<<3);
  LEDS&=~(1<<4);
  LEDS&=~(1<<5);
  LEDS&=~(1<<6);
  LEDS&=~(1<<7);
  LEDS&=~(1<<8);
  LEDS&=~(1<<9);
  LEDS&=~(1<<10);
  LEDS&=~(1<<11);
  LEDS&=~(1<<12);
  LEDS&=~(1<<13);
  LEDS&=~(1<<14);
  LEDS&=~(1<<15);

  //Test Print
```

```
xil_printf("%u \n", strength_avg);

//Check for large metal detection
if (large_threshold<strength_avg)
{
  data[0] = total_cnt;
  data[1] = small_cnt;
  data[2] = med_cnt;
  data[3] = large_cnt;
  state_cnt_large++;
}
//Check debounce metal detection count

else if (med_threshold<strength_avg)
{
  data[0] = total_cnt;
  data[1] = small_cnt;
  data[2] = med_cnt;
  data[3] = large_cnt;
  state_cnt_med++;
}
else if (small_threshold<strength_avg)
{
  data[0] = total_cnt;
  data[1] = small_cnt;
  data[2] = med_cnt;
  data[3] = large_cnt;

  state_cnt_small++;
}
else
{
  data[0] = total_cnt;
  data[1] = small_cnt;
  data[2] = med_cnt;
  data[3] = large_cnt;
}
if(strength_avg < nothing_threshold && (state_cnt_large>=1000))
{
  total_cnt++;
  large_cnt++;
  state_cnt_small=0;
  state_cnt_med=0;
  state_cnt_large=0;
}
```

```
if(strength_avg < nothing_threshold && (state_cnt_med>=1000))
{
  total_cnt++;
  med_cnt++;
  state_cnt_small=0;
  state_cnt_med=0;
  state_cnt_large=0;
}
if(strength_avg < nothing_threshold && (state_cnt_small>=1000))
{
  total_cnt++;
  small_cnt++;
  state_cnt_small=0;
  state_cnt_med=0;
  state_cnt_large=0;

}

//Display counts
seg_disp(data);

//Strength Meter
if(strength_avg>=30000)
{
  LEDS |= (1>>0);
}
if(strength_avg>=30500)
{
  LEDS |= (1<<1);
}
if(strength_avg>=31000)
{
  LEDS |= (1<<2);
}
if(strength_avg>=31500)
{
  LEDS |= (1<<3);
}
if(strength_avg>=32000)
{
  LEDS |= (1<<4);
}
if(strength_avg>=32500)
{
  LEDS |= (1<<5);
```

```
    }
    if(strength_avg>=33000)
    {
      LEDS |= (1<<6);
    }
    if(strength_avg>=33500)
    {
      LEDS |= (1<<7);
    }
    if(strength_avg>=34000)
    {
      LEDS |= (1<<8);
    }
    if(strength_avg>=34500)
    {
      LEDS |= (1<<9);
    }
    if(strength_avg>=35000)
    {
      LEDS |= (1<<10);
    }
    if(strength_avg>=62000)
    {
      LEDS |= (1<<11);
    }
    if(strength_avg>=64000)
    {
      LEDS |= (1<<12);
    }
    if(strength_avg>=65000)
    {
      LEDS |= (1<<13);
    }

  }
  return 0;
}
```