Abstract

The project consists of designing and building a system that monitors the phase of two pulses A and B. Three colored LEDs are used to identify the phase comparison. When the rising edge of pulse A occurs 10 us or more before the rising edge of pulse B, a red LED is on. If the rising edge of pulse B occurs 10 us or more before the rising edge of A, a yellow LED is on. If the rising edge of A and B are within 10 us of each other, a green LED is on. This project uses the 8-bit flash microcontroller ATMEL AT89C51RD2.

Components:

(1) 8-bit flash microcontroller ATMEL AT89C51RD2
(1) HD44780U HITACHI Dot Matrix Liquid Crystal Display
(3) LEDs
(1) TC4584BP Hex Schmitt Trigger
(1) On/Off switch
(3) 300Ω Resistor
(1) 0.001 µF Capacitor
(1) Potentiometer

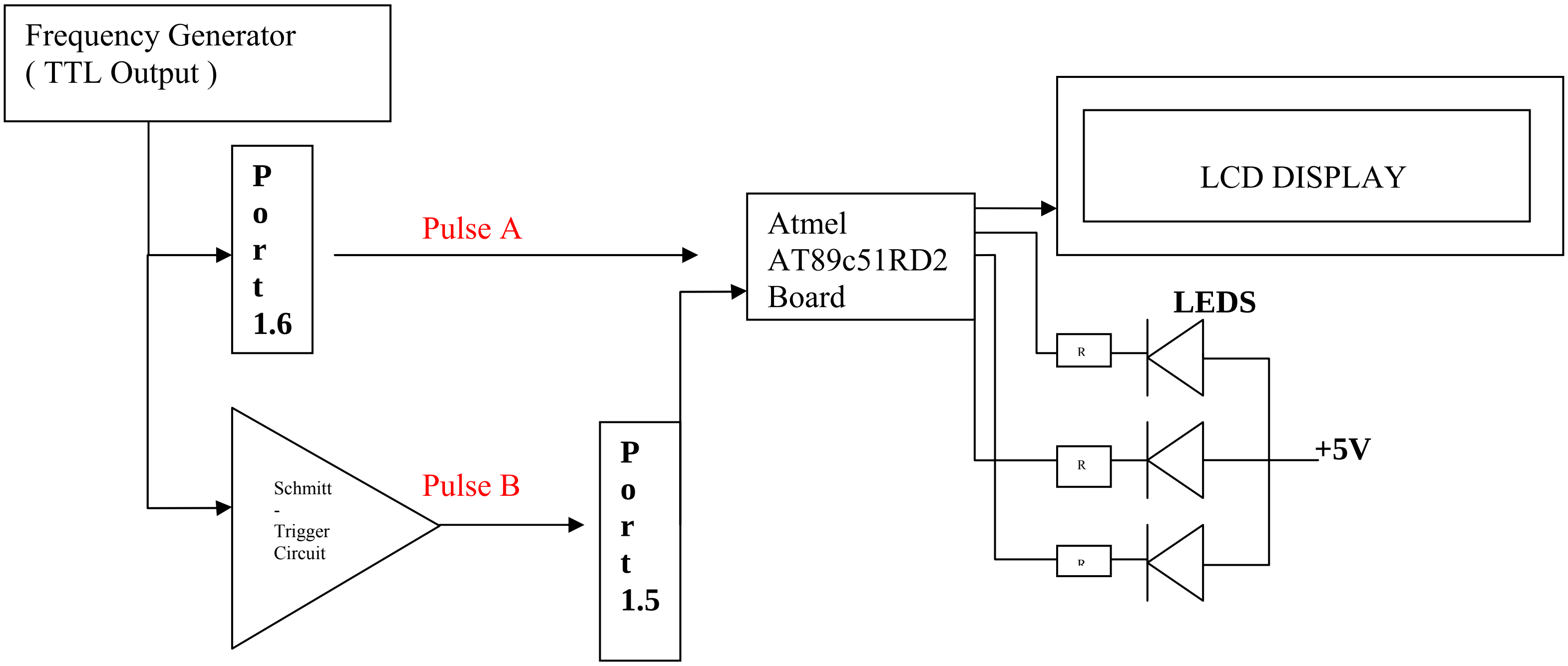


Figure 1: Connection Diagram **(Overview)**

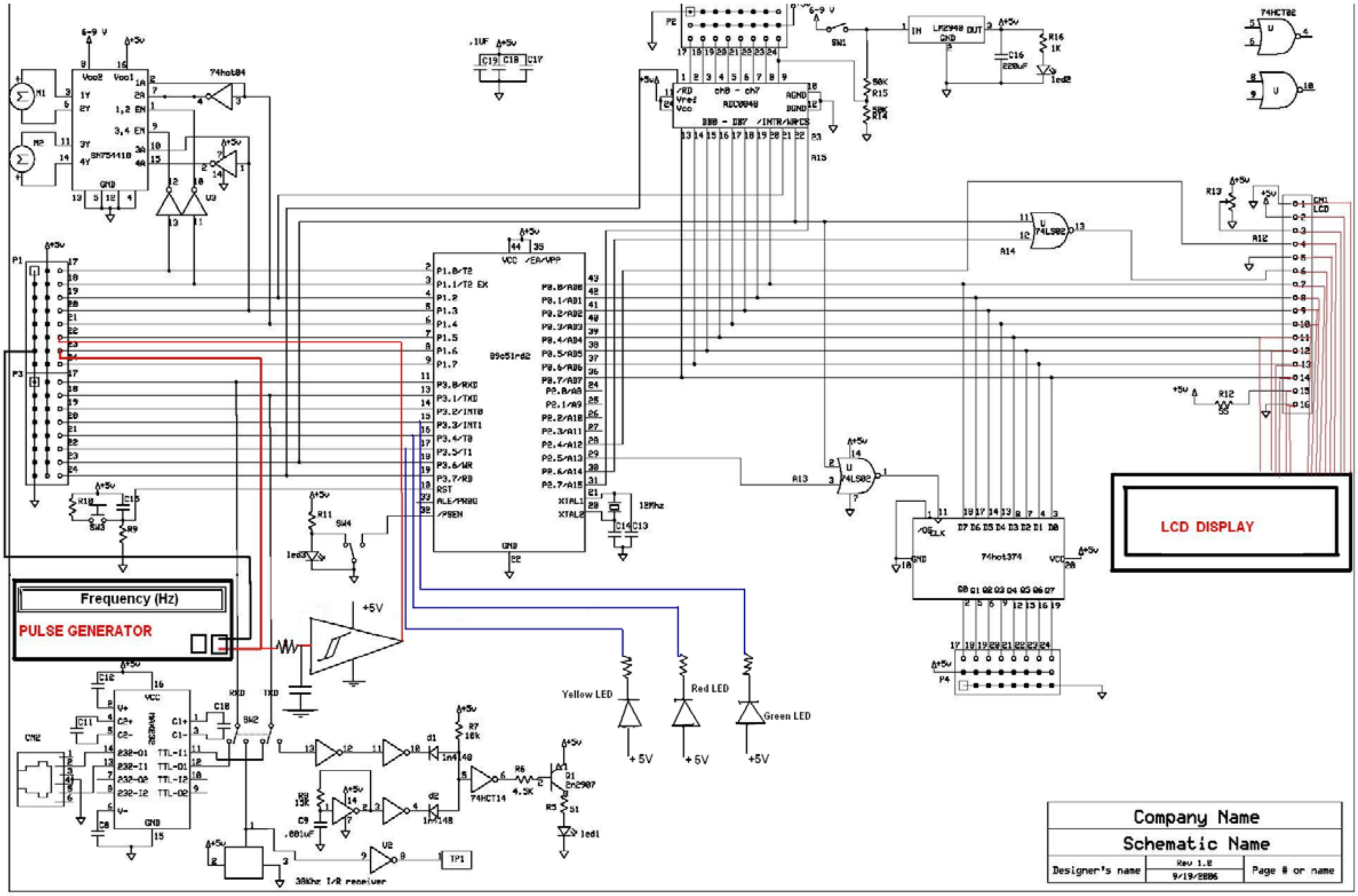


Figure 1: Detailed Circuit Diagram

The idea in designing this device is to use the 8-bit flash microcontroller ATMEL AT89C51RD2 to compare the rising edge time of pulse A and pulse B. The PCA (Programmable Counter Array) included in this processor consists of a dedicated timer/counter which serves as the time base for an array of five compare/capture modules. In this project, I used the module 3 and 2 respectively for the pulses A and B. Both modules are set put in rising edge capture mode and interrupt enabled.

I chose the rising edge capture mode because my design needs to capture the rising edge of each pulse in order to calculate the phase difference.

Source Code Documentation:

Description of the algorithms

The software makes use of 2 modules: module 3 and module 2.
The project uses an interrupt routine that measures the rising edge time of pulses A and B and calculates the phase difference between the pulses coming at ports P1.6 and P1.5. Two flags are used to determine the first rising edge between pulse A and pulse B. The external variables rising_flag and difference are global variables known to the main code. In the main code, the pins P1.3, P1.4 or P1.5 are set according to the value of the difference and the phase difference is sent to an LCD display.

- **Interrupt routine design:**

```
1 – If the PCA overflow flag CF (in the CCON SFR) is set 1 when the PCA timer overflows.
        1.1 – Reset the flag CF
        1.2- Increment the PCA overflow variable

2- Otherwise must be a capture
        2.1- If rising edge of pulse A
                2.1.1- Get the rising edge time with CCAP3
                2.1.2- Reset CCF3 for the next interrupt
                2.1.3- Check if pulse A rising edge occurs first and set a flag

        2.2- If rising edge of pulse B
                2.2.1- Get the rising edge time with CCAP2
                2.1.2- Reset CCF2 for the next interrupt
                2.1.3- Check if pulse B rising edge occurs first and set a flag


        2.3- If pulse A rising edge occurs first
                2.3.1- Set the rising flag A to 1
                2.3.2- Calculate the difference between the rising edge times.

        2.4- Otherwise if pulse B rising edge occurs first
                2.4.1- Set the rising flag B to 1
                2.4.2- Calculate the difference between the rising edge times.
```

- **Initialize routine design**

```
1- Initialize the display
2- Initialize the PCA
    2.1- Put PCA in osc divide by 6 mode and enable PCA interrupt for overflow
    counting
    2.2- Put module 3 in rising and falling edge capture mode and enable interrupt
    2.3- Put module 2 in rising and falling edge capture mode and enable interrupt
    2.4- Enable PCA clock source (PCA starts counting)
    2.5- Interrupt Enable Control 0
```

- **Main program design**

```
1- Call the Initialize() routine
2 – While loop
    2.1- If the phase difference is greater than 10 and the rising flag A is 1
        2.1.1- Activate pin P1.4
        2.1.2- Display message
    2.2- If the phase difference is greater than 10 and the rising flag B is 1
        2.2.1- Activate pin P1.5
        2.2.2- Display message
    2.3- If the phase difference is less than 10
        2.3.1- Activate pin P1.3
        2.3.2- Display message
```

- **Display routine design**

```
1- LCD Delay module
2- LCD Driver module
3- Display text module
4- Display number module
```

How does the device work internally?

The system has:

- 2 pulse signal inputs:
  Pulse A input at port P1.6
  Pulse B input at port P1.5

- 3 output pins:
  P1.3 connected to the Green LED
  P1.4 connected to the Red LED
  P1.5 connected to the Yellow LED

The device uses a single pulse coming from the TTL output of a pulse generator. Pulse A is the initial pulse. This initial pulse is inputted to a TC4584BP Hex Schmitt Trigger that generates pulse B. The pulses A and B are inputted to Ports P1.6 and P1.5 respectively and the processor calculates the phase difference between the two pulses. According to the phase difference obtained, the pin P1.3, P1.4 or P1.5 is activated and turns ON one of the colored LEDs

Setup of the TC4584B Hex Schmitt Trigger

In order to meet the requirements of the project, two pulses A and B need to be generated. The generation of pulse B is achieved by connecting a single resistor-capacitor network to a Schmitt trigger. The capacitor connects between the input and ground and the resistor connects between the input and the pulse generator. The output is a continuous pulse whose characteristics depends on the values of R and C, and the threshold points of the Schmitt trigger.

From the characteristics of the TC4584B Hex Schmitt Trigger:

$$\rightarrow \frac{1}{R.C} = \frac{1}{t}$$

I decided to use a potentiometer to get different delays between the pulses A and B. By adjusting the knob of the potentiometer with a screwdriver, I can change the RC value which automatically changes the delay between pulse B and pulse A.

These are some delays obtained by adjusting the potentiometer resistance value:

For C = 0.001µF,
When R = 10kΩ, delay t = 10µs, the Green LED is ON
When R = 20kΩ, delay t = 20µs, either the Red or Yellow LED ON

Physical value of the capacitor: C = 0.001 µF
Physical characteristic of the Potentiometer: $0 \leq R \leq 100$ kΩ

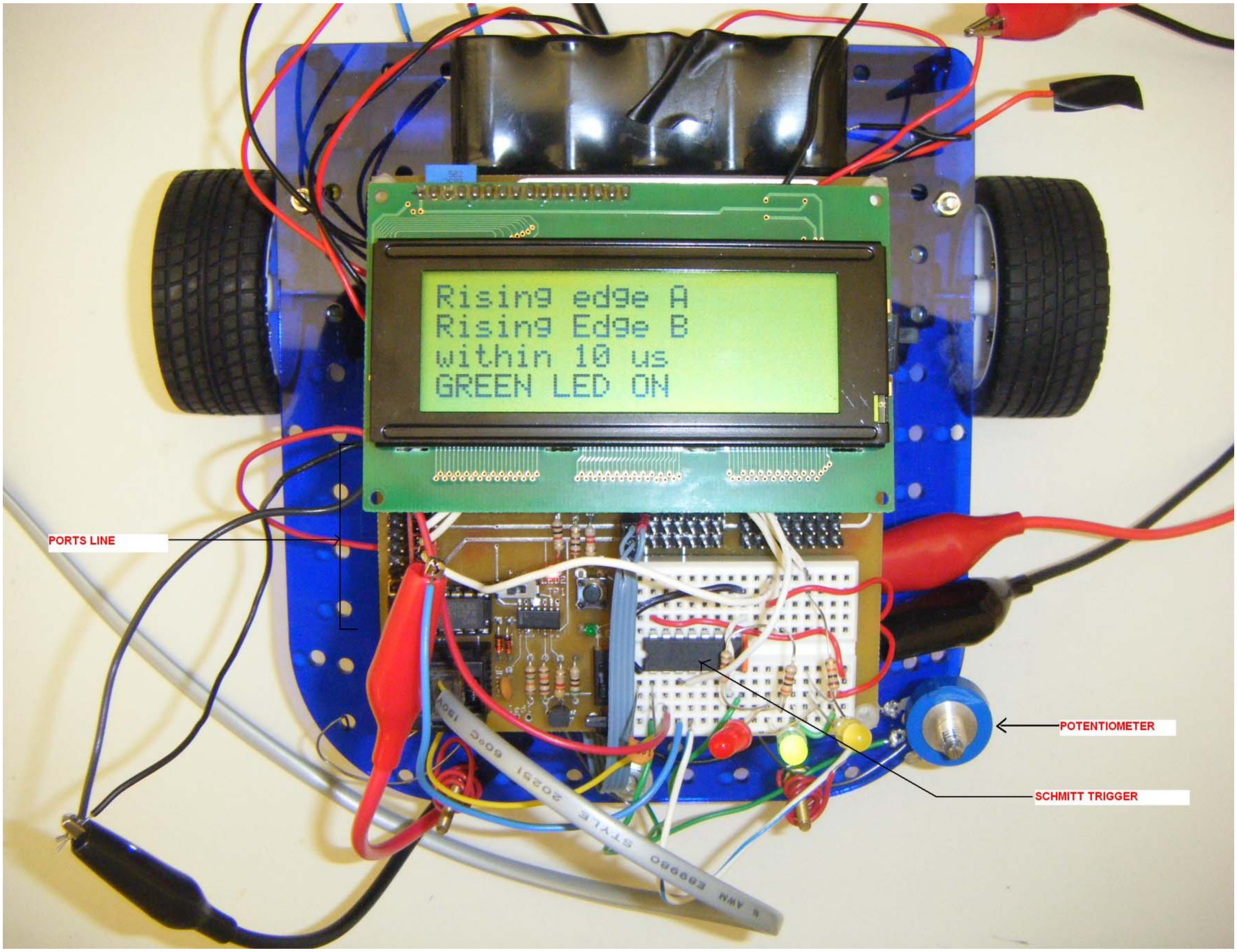
Extra features

- The device is portable, using a pack of 4 x 1.5V AA batteries.
- An On/Off switch is added to easily power the device on and off.
- An HD44780U HITACHI LCD is added to display the leading pulse and the state of the LED.

Reference used:

Microelectronic Circuit Analysis and Design, (third edition)
Author: Donald Neamen.
Publisher: McGraw-Hill Science
ISBN-13: 978-0073285962
Chapter 15, page 1078 to 1082.

TOSHIBA TC4584BP Hex Schmitt Trigger datasheet from: http://eecs.evansville.edu
8-bit flash microcontroller ATMEL AT89C51RD2 datasheet



```
Rising edge A
Rising Edge B
within 10 us
GREEN LED ON
```

Final Board Picture

Conclusion

The main problem encountered with this project was to calculate the delay between the rising edge time of pulses A and B in the interrupt routine code. In hardware, I first used a fixed value resistor of 10kΩ, with resulted in a fixed delay between pulse A and B. Therefore, I decided to use a potentiometer that allowed the Schmitt trigger to vary the delay efficiently, just by adjusting the potentiometer knob with a screwdriver. An LCD has been added to the board to display additional information on the phase comparison and the state of the LEDs (See final picture). Finally, the device had been tested multiple times and works successfully.