Thomas Driscoll (tfd2)
Kelvin Ma (kelvinm2)
Rishabh (rishabh3)

ECE 110 (H) FINAL PROJECT LAB REPORT

Introduction

The goal of our project was to design a remote controlled bicycle lock that would allow bicyclists on campus to have faster and more security with their bikes. In the modern age, the group felt that it is imperative to update the average bike lock with this system. The benefits of this product will be that the locks will be easier to use, more secure against lock-picking thieves and an overall better safety mechanism.

Design description

The device was originally designed to have have two pieces:

- A handheld remote control, goes with the user
- A physical lock, stays on the bike

The remote will be used to enter a binary sequence, which will be transmitted to the lock, which will verify the sequence internally and lock and unlock accordingly.

Since a physical remote and lock were a stretch for the project, we built them into the circuit. The remote still contains the binary sequence, which was transmitted to the circuit and represented the locking of the bike through a series of LEDs.

How it works

Remote

The remote will have three **buttons**, which will qualify as our **sensors**.

Button 0	Binary key 0
Button 1	Binary key 1
Button 2	Code reset

The remote will use IR signals to communicate with the lock controller, transmitting the keypresses synchronously.

Lock device

The lock controller has IR sensors which will pick up on the signals sent by the built-in remote. As the controller receives each bit from the remote, it compares it with the corresponding bit in its internally stored keycode. For this project, the keycode will probably be 8 bits long, though this can be arbitrarily scaled. Once the controller has received 8 bits of valid input, it will open the lock mechanism. If it unlocks, or receives a Code reset signal, it will return to the beginning of the 8-bit sequence and continue listening.

Analysis of components

IR Transmitter/Receiver Pair:
We used a pair of side looking IR transmitters and receivers to transmit the data from the remote to the authentication system in order to unlock the system. The IR pair operates at a frequency of 940nm, and the detector is an NPN transistor that is biased by incoming IR light which causes a change in the voltage across the detector (Letting all of it go through while not receiving a signal and limiting it when light is shone upon it). The emitter can be driven up to 50mA which was controlled by using a suitable resistor. Using the properties of this transmitter receiver pair, we parsed information to our decoding circuit which would treat a long pulse as a high and a short pulse as a low.

7 Seg Decoder
We used a 7 segment decoder to display the amount of bits that had been parsed to the decoder circuit in order to keep track of them. The display is implemented by setting different pins to and high and low, each corresponding to an LED on the display in order to display a number.

Comparator
We used a comparator to compare the two voltages in the circuit in order to determine that the short burst received was a “low” and that the long burst was a “high” information which was parsed to the next part of the circuit to authenticate the sequence.

Shift Registers
We used shift registers (a series of delay stage flip-flops) these microcontrollers were used to convert the serial data into a parallel form to be compared by the comparator in order to determine what was a high and what was a low.

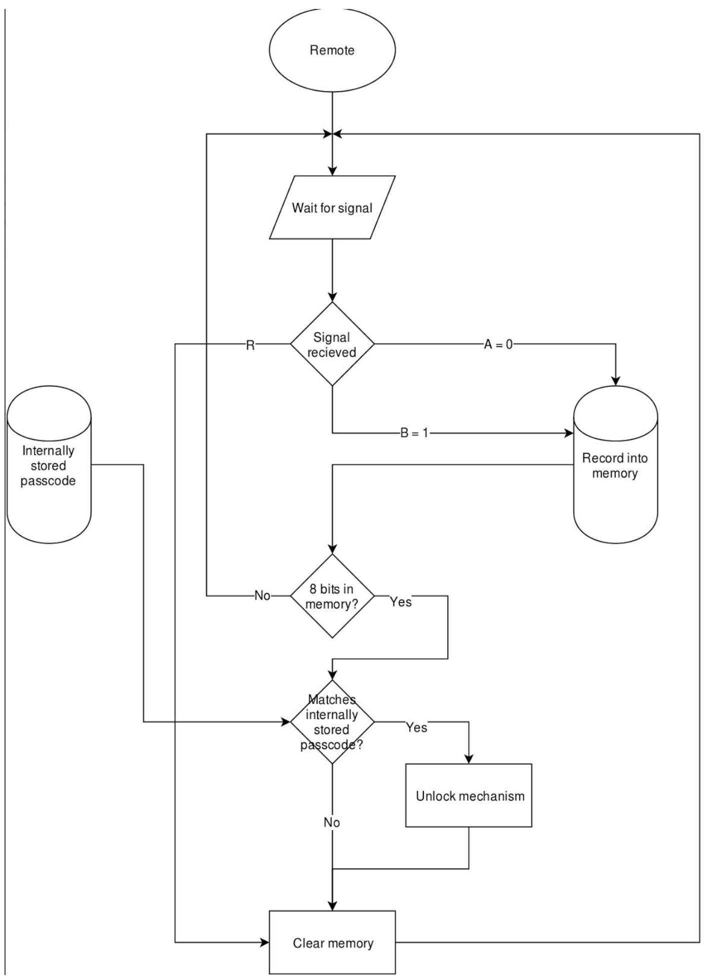
Logic Gate Chips - NAND and NOR
We used these gates as part of the logic circuit to convert the linear band transmission into a binary equivalent.

Binary Counter
The binary counter was the most integral part of the conversion logic circuit which set the short bursts and long bursts to highs and lows by comparing them to the clock.

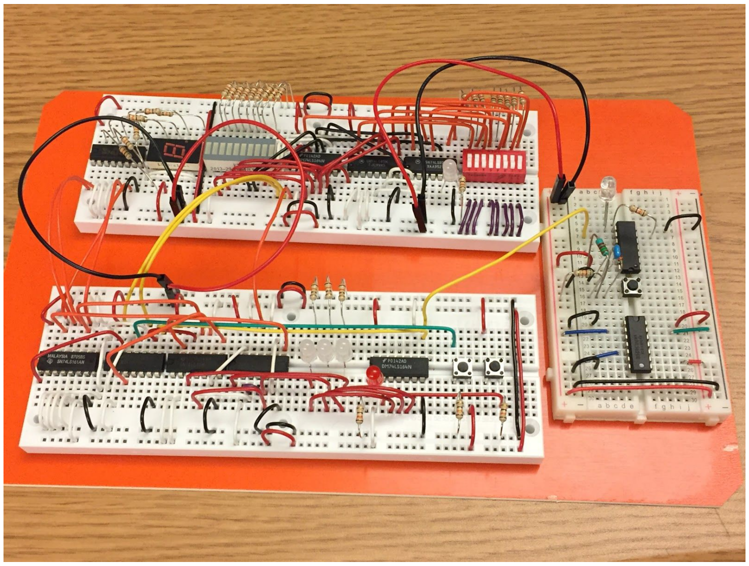
Physical Switch Block
Finally we used a physical switch block with 8 switches each representing a bit in the circuit to which the logic circuit passed highs and lows along the individual switches, which if correct would light up the LED representing the execution of the unlocking mechanism.

Arduino PWM
We used an arduino to drive a schmitt inverter across the IR detector acting as the comparative clock for determining whether it was a short burst or long burst.

Circuit Schematic



Functional schematic of how the remote controlled bike lock processes inputs to unlock the lock mechanism.



The breadboard prototype uses digital logic DIP chips to implement the signal decoding system which translates a single-band transmission input into a binary sequence of 1’s and 0’s.

Conclusion

Being the way it was, our project was pretty complicated right from the start. The most important part of the project was the authentication system and parsing information from the IR LEDs to the authentication system. After initial debate on whether we should use an Arduino for the authentication system or to build one completely on our own, we decided to use the logic circuit favouring it over the arduino for more accuracy in processing. Apart from the logic gate construct itself, we initially decided to use a normal clock IC for the comparison circuit, however it was producing a very noisy signal even after using a schmitt inverter and getting the transmission to match it would have been very difficult. Due to these reasons, we decided to use an Arduino to drive the clock, using a simple PWM code which led to the conclusion of our work on the project. A switch was used to simulate what the IR detector would do in terms of high and low. We were able to overcome these challenges, producing a successful final project which performed excellently. Due to the difficulties of the project, we were expecting to not be able to complete this project on time. However, after meeting outside the lab we were able to get the project to work. Going forward we would look towards reducing the circuit’s footprint and try and get rid of the Arduino. We would then add the unlocking mechanism which would work either by using a motor or electromagnets to produce a final product.