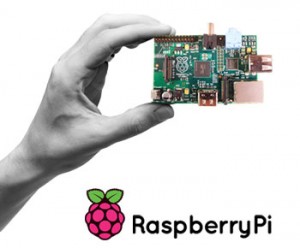
Implementing Binary Counting

Marlon Myers

Computer Hardware Systems: EMT 2370

Instructor: Prof. Carranza

|  |  |
| --- | --- |
| Table of Contents | Page |
| Objective | 3 |
| Introduction | 4 |
| Equipment | 5 |
| Discussion | 6 |
| Conclusion | 8 |
| Reference | 9 |

Objective:

* To implement a Binary Counting sequence, using 7476 Dual JK Flip-Flop chips, LED lights and a Raspberry Pie.
* To use C Programming Language to provide a clock impulse to the JK Flip-Flop chip, to either Increment or Decrement between 0 – 15 in binary numbers.

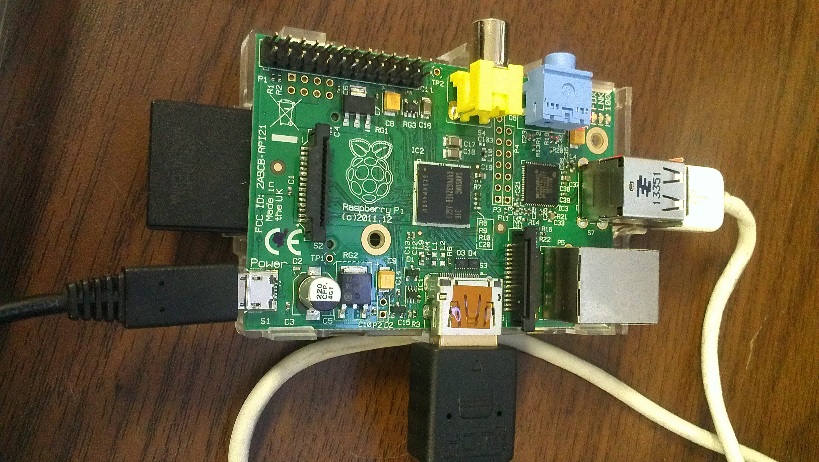
Introduction:

The purpose of this project is to display one of many uses of a Raspberry Pi. A Raspberry Pi is a low cost, credit-card sized computer that plugs into a computer monitor or TV, and uses a standard keyboard and mouse. It is capable of doing everything you’d expect a desktop computer to do, from browsing the internet and playing high-definition video, to making spreadsheets, word-processing, and playing games. In this Project, the Raspberry Pi is operated by the Noobs operating system; New Out Of the Box Software. Within the Noobs operating system, IDLE (Integrated Development Environment) is used as the default implementation of the C programming language. Connected to the Raspberry pi, is a circuit constructed to display a Binary Counting sequence. The word binary comes from "Bi-" meaning two. We see "bi-" in words such as "bicycle" (two wheels) or "binocular" (two eyes). A Binary Number is made up of only 0s and 1s. A collective use of 4 LED lights are used to implement a Binary counting sequence from 0 to 15. The circuit uses JK Flip-Flop chips; 7476 chips, to allow the circuit to count in a binary sequence.

Equipment:

* Raspberry Pi
* 3 Alligator Clip
* 4 LED’s
* 2 JK Flip-Flop (7476) chips
* 3 Resistors (330 ohms)
* Jumper Wires
* Monitor
* Keyboard
* Mouse
* Power source (Micro-USB Cord)
* HDMI Cord and Converter (VGA)
* 2 Bread-Board

Discussion:

1. To begin the experimentation, the Raspberry Pi is connected to the JK Flip-Flop circuit. As seen below



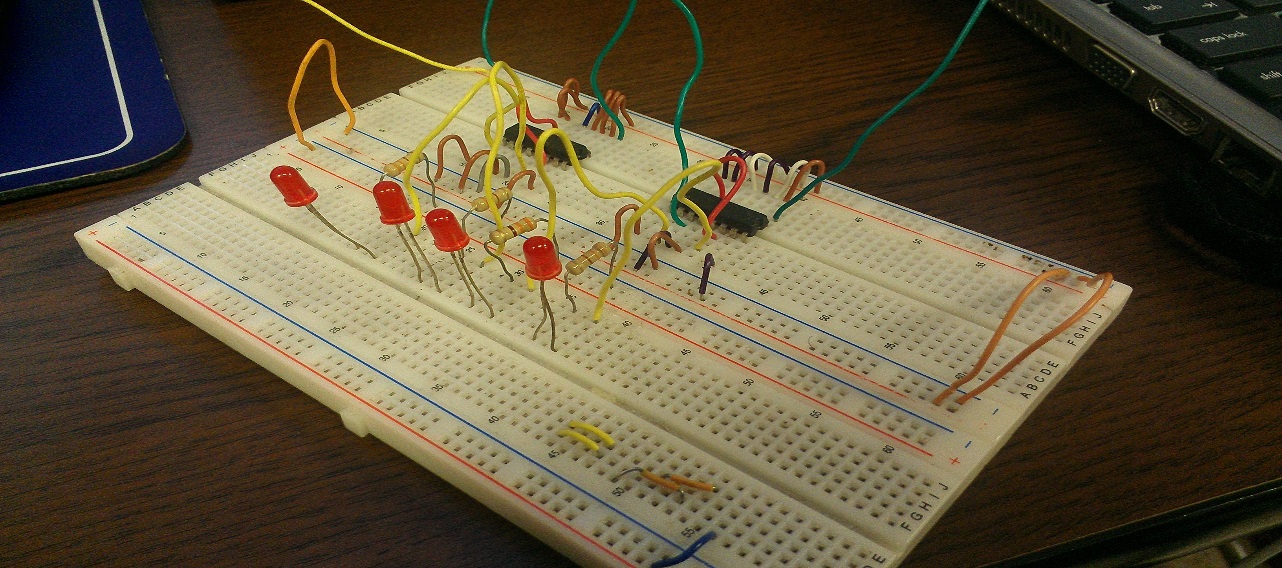
1. Within the Noobs Operating System on the Raspberry Pi, an application called IDLE is used to Run C programming language. The C language is used to perform outputs on the Bread Board through LED lights.

|  |
| --- |
| Import RPi.GPIO as GPIO //import the general purpose input output system on (RPi)  Import time //import the use of time  GPIO.setmode(GPIO.BOARD) //import general purpose input out output on Breadboard  GPIO.setup(7,GPIO.OUT) //set the output pin to be Pin 7 on RPi  For x in range (0,32): //the amount of times the code will repeat/loop  GPIO.OUTPUT(7,true) //turning on or giving clock pulse to the output of pin 7  time.sleep(0.5) //turns on for 0.5 seconds  GPIO.OUTPUT(7,False) //turning of or giving no clock pulse to the output pin 7  time.sleep(0.5) //turns off for 0.5 seconds  GPIO.cleanup() |

Note: GIOP = General Purpose Input Output

RPi = Raspberry Pi

1. Clock pulses from the Raspberry Pi, to the output of pin 7; which is connected to the first input clock pulse of JK Flip-Flop along the Bread Board allows the LED lights to count in binary sequence from 0 to 15, as seen below.

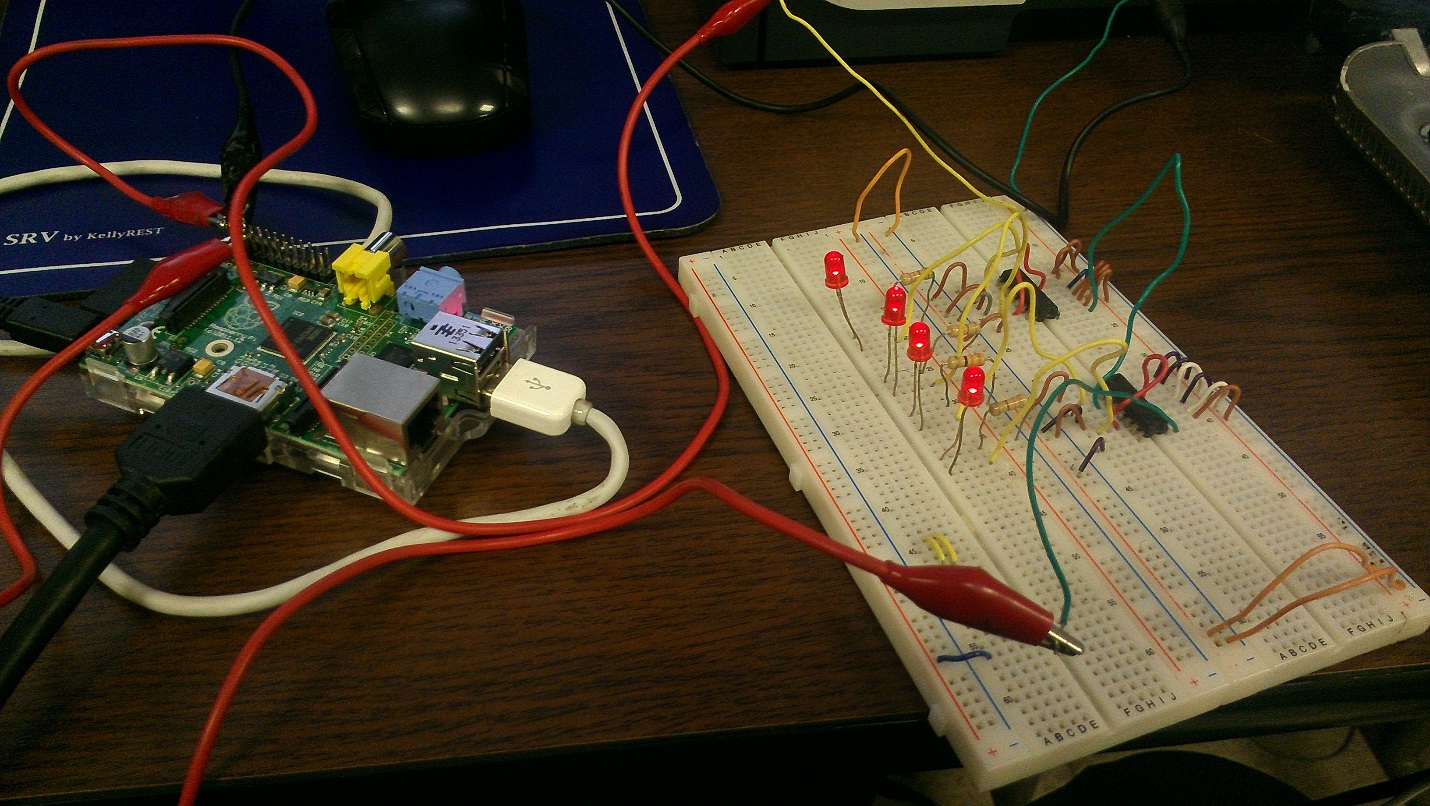


1. The table below displays the Decimal Converted value of the binary number implemented on the Bread Board.

|  |  |
| --- | --- |
| Binary Numbers | Decimal Numbers |
| 0000 | 1 |
| 0001 | 2 |
| 0010 | 3 |
| 0011 | 4 |
| 0100 | 5 |
| 0101 | 6 |
| 0110 | 7 |
| 0111 | 8 |
| 1001 | 9 |
| 1010 | 10 |
| 1011 | 11 |
| 1100 | 12 |
| 1101 | 13 |
| 1110 | 14 |
| 1111 | 15 |

Conclusion:

In conclusion, my implementation of a Binary circuit on a bread board was a success. I was able to control the circuit on the Bread Board with a Raspberry Pi; Micro-Computer, which was implemented with C programming language. The LED lights light up in a binary sequence form 0 to 15.



References:

http://www.raspberrypi.org

http://www.mathsisfun.com/binary-number-system.html