RISC-V TALENT DEVELOPMENT PROGRAM - 2025

Powered by Samsung Semiconductor India Research(**SSIR)** along with VLSI System Design(**VSD)**

**TITLE OF PROJECT : An automatic light system**

Submitted by,

**Shivabasavesh A S**

3rd Year ECE

THE NATIONAL INSTITUTE OF ENGINEERING MYSURE

**Course Instructor**

Kunal Ghosh

CO-FOUNDER OF VLSI SYSTEM DESIGN(**VSD)**

# Contents

|  |  |
| --- | --- |
| 1. | Overview |
| 2. | Components Required |
| 3. | Logical Diagram and Expressions |
| 4. | Circuit Connection |

**Overview;**

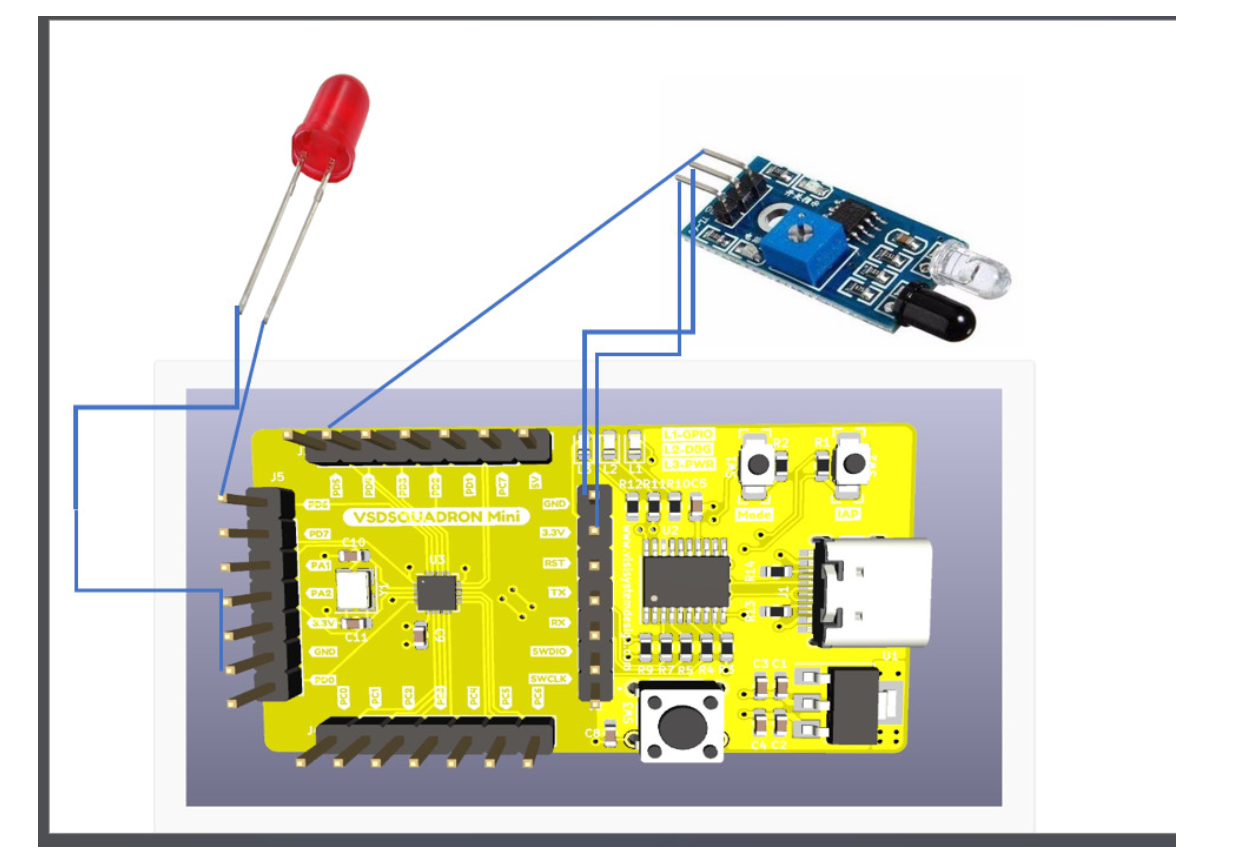
An automatic light system is a setup designed to automatically control the lighting based on the presence or absence of individuals within its detection range. This system will also give the indication of motion detected by blinking the led 3 times.

**Components Required**

* VSDSquadron Mini Board
* IR Sensor
* LEDs
* Bread Board
* USB Cable
* Jumper Wires

**Pin Diagram**

| **IR SENSOR, LED** | **VSD SQUADRON BOARD** |
| --- | --- |
| VCC OF IR | 3.2V |
| GND OF IR | GND |
| OUT OF IR | PIN 4 |
| LED | PIN 6 |

****

**Working**

* The IR sensor is strategically placed in a location where it can detect the movement of individuals within its sensing range.
* The IR sensor continuously monitors its surroundings for any changes in infrared radiation caused by the movement of individuals.
* When someone enters the detection range of the IR sensor, it detects the change in radiation and triggers an output signal.
* When the IR sensor detects motion, it sends a signal to the microcontroller which then activates the LED lighting system. The LED lights up, providing illumination and the led will blink 3 times in the area where motion is detected which gives indication of motion detected.

**C CODE**

**//These include the necessary header files (ch32v00x.h and debug.h) for the CH32V microcontroller and debugging purposes.**

**#include <ch32v00x.h>**

**#include <debug.h>**

**//pin configuration**

**void GPIO\_Config(void)**

**{**

**GPIO\_InitTypeDef GPIO\_InitStructure = {0}; //structure variable GPIO\_InitStructure of type GPIO\_InitTypeDef which is used for GPIO configuration.**

**RCC\_APB2PeriphClockCmd(RCC\_APB2Periph\_GPIOD, ENABLE); // to Enable the clock for Port D**

**//pin 4 OUT PIN FOR IR SENSOR**

**GPIO\_InitStructure.GPIO\_Pin = GPIO\_Pin\_4 ; // Defines which Pin to configure**

**GPIO\_InitStructure.GPIO\_Mode = GPIO\_Mode\_IPU; // Defines Output Type**

**GPIO\_Init(GPIOD, &GPIO\_InitStructure);**

**//pin 6 IS LED PIN**

**GPIO\_InitStructure.GPIO\_Pin = GPIO\_Pin\_6 ; //**

**GPIO\_InitStructure.GPIO\_Mode = GPIO\_Mode\_Out\_PP; // Defines Output Type**

**GPIO\_InitStructure.GPIO\_Speed = GPIO\_Speed\_50MHz; // Defines speed**

**GPIO\_Init(GPIOD, &GPIO\_InitStructure);**

**}**

**//main function**

**int main(void)**

**{**

**uint8\_t IR = 0;**

**uint8\_t set=1;**

**uint8\_t reset=0;**

**uint8\_t a=0;**

**NVIC\_PriorityGroupConfig(NVIC\_PriorityGroup\_2);// Configuring NVIC priority group**

**SystemCoreClockUpdate();// Update System Core Clock**

**Delay\_Init();//Initialize Delay**

**GPIO\_Config();//Call GPIO configuration function**

**while(1)**

**{**

**IR = GPIO\_ReadInputDataBit(GPIOD, GPIO\_Pin\_4);**

**if (IR==1)//Read state of Pin 4 (IR sensor)**

**{ // for blinking of led three times upon motion detection**

**for(a=0;a<3;a++){**

**GPIO\_WriteBit(GPIOD, GPIO\_Pin\_6, set);**

**Delay\_Ms(200);**

**GPIO\_WriteBit(GPIOD, GPIO\_Pin\_6,reset);**

**Delay\_Ms(100);}**

**}**

**}**

**}**

**Applications**

* **Security Lighting** : These systems can be used for security lighting in outdoor spaces, such as gardens, driveways, and pathways, to deter intruders and provide visibility at night.
* **Home Automation** : Automatic light systems can be installed in homes, particularly in areas such as hallways, staircases, and bathrooms, where lights need to be turned on/off based on occupancy.
* **Energy Efficiency** : Automatic light systems contribute to energy conservation by ensuring that lights are not left on unnecessarily when the area is unoccupied.
* **Accessibility** : These systems can improve accessibility for individuals with disabilities by providing automatic illumination in response to their movement.

**Conclusion**

During the VSD Squadron mini Internship, I embarked on a journey exploring various aspects of VLSI system design on the RISC-V architecture, alongside open-source EDA tools.

**Installation and Setup :** I began by setting up the development environment, including installing VirtualBox for running Ubuntu and configuring the RISC-V toolchain, Yosys, Icarus Verilog, and GTKWave.

**Understanding RISC-V Architecture :**I delved into the fundamentals of the RISC-V architecture, learning about its open, versatile instruction set, instruction formats, operand types, and various instruction types and their opcodes.