techno india university
WEST BENGAL

IIT Bombay eYantra
Engineering a better tomorrow

Design a project to toggle the status of 2 Bar-graph LEDs depending on whether the Interrupt Switch is pressed or released.

By

SOUVIK PAL (ID:211001002042 BATCH:BEC3 YEAR:3RD)

SOUNABHA GHOSH (ID:221001012004 BATCH:BEC3 YEAR:3RD)

TURYA DEY (ID:211001003006 BATCH:BEE3 YEAR:3RD)

AYAAN NANDI (ID:211001001002 BATCH:BCS3A YEAR:3RD)

CHIRAG CHAKRABORTY (ID: 211001001231 BATCH:BCS3C YEAR:3RD)

RAHUL BANARJEE (ID:181001021009 BATCH:BCS4C YEAR:4TH)

techno india university
WEST BENGAL

IIT Bombay eYantra
Engineering a better tomorrow

Table of Contents

Introduction on Atmega2560:

The heart of our project is ATmega2560 microcontroller. The ATmega2560 is high-performance, low-power Microchip 8-bit AVR® RISC-based microcontroller combines 256 KB ISP flash memory, 8 KB SRAM, 4 KB EEPROM, 86 general purpose I/O lines, 32 general purpose working registers, real-time counter, six flexible timer/counters with compare modes, PWM, four USARTs, byte-oriented Two-Wire serial interface, 16-channel 10-bit A/D converter, and a JTAG interface for on-chip debugging. The device achieves a throughput of 16 MIPS at 16 MHz and operates between 4.5-5.5 volts. It executes instructions in a single clock cycle, allowing it to balance power consumption and processing speed. It has an operating voltage of 5 volts, and a recommended input voltage of 7–12 volts. The board can be powered by a USB cable, the board's VIN, or a power jack or battery.

Accessing Ports:

To access the ports of the ATmega2560 we need associated registers which control the behavior of the port. When the ports are used as GPIO there are 3 registers that we need to access, which are:

1. DDRx 2. PORTx 3. PINx

Register Structures:

1. *Data Direction Register for Port J and Port K (DDrX):*
   - DDrJ and DDrK are 8-bit registers where each bit corresponds to a pin in PORTJ and PORTK.
   - Setting a bit to 1 configures the corresponding pin as an output, and setting it to 0 configures it as an input.
2. *Port J and Port K Output Register (PORTX):*
   - PORTJ and PORTK are 8-bit register where each bit corresponds to a pin in PORTJ and PORTK.
   - Writing a 1 to a bit sets the corresponding pin high (if configured as an output), and writing a 0 sets it low.
3. *Port E Input Register (PINE):*
   - PINE is an 8-bit register where each bit corresponds to a pin in PORTE.
   - Reading a bit in this register gives the current logic level on the corresponding pin.

Aim of project:

The aim of the project, "Toggle LED Bar-graphs with Interrupt Switch," is to demonstrate the integration of a microcontroller, specifically the ATmega2560, in a system that responds to external events through the use of interrupts. The primary goals of the project include:

Interrupt-driven System:

- Showcase the utilization of external interrupts on the ATmega2560 microcontroller.
- Use the INT0 external interrupt to detect state changes in a connected switch.

LED Control:

- Interface with output devices, in this case, Bar-graph LEDs connected to PORTJ.
- Toggle the status of two Bar-graph LEDs based on the interrupt switch's press or release events.

Understanding AVR Ports and Registers:

- Gain practical experience in configuring and manipulating I/O ports on the ATmega2560.
- Interact with Data Direction Registers (DDRx), Output Registers (PORTx), and Input Registers (PINx) to control the state and direction of port pins.

Learning Embedded System Concepts:

- Provide a hands-on experience in embedded systems programming.
- Understand the importance of interrupts for real-time event handling in microcontroller applications.

Educational Purpose:

- Serve as a learning tool for individuals interested in microcontroller programming and embedded systems.
- Offer insights into addressing ports and interrupts.

Materials/software, programming languages used:

Materials:

1. ATmega2560 Microcontroller: The primary component, serving as the brain of the project.
2. Bar-graph LEDs: Two Bar-graph LEDs are used as visual indicators.
3. Interrupt Switch: A push-button switch is used to trigger interrupts.
4. Power Supply: A 5V power supply to power the ATmega2560.
5. Ground Connection: Ground is a reference point for voltage in the circuit.

Software:

1. Microchip Studio: An integrated development environment (IDE) for AVR microcontrollers, including the ATmega2560.
2. AVR-GCC Compiler: The GNU Compiler Collection for AVR, used for compiling the C code.
3. Proteus: A simulation software to model and test the project virtually before implementing it on hardware.

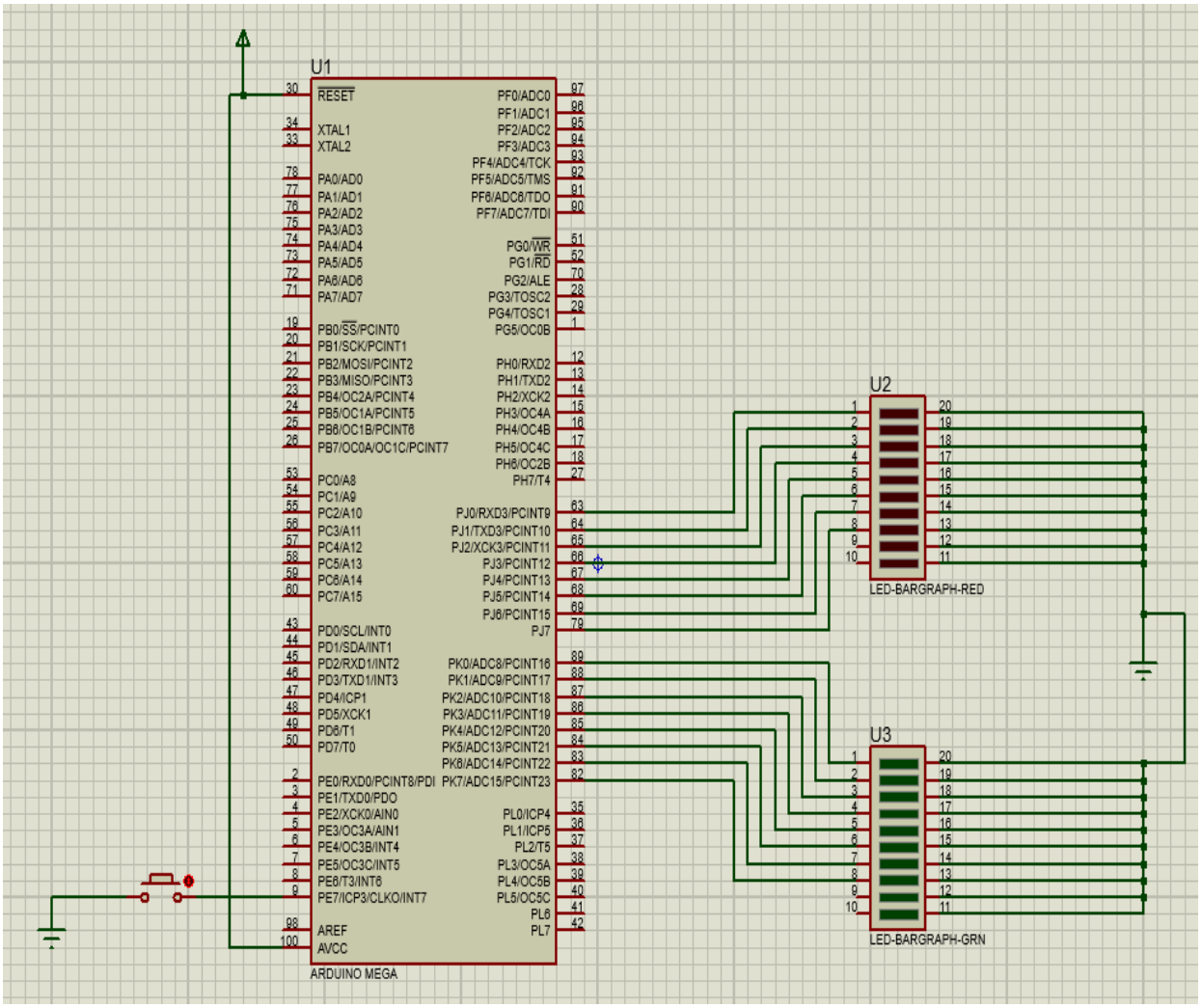
Programming Language:

The project is implemented using the C programming language. C is widely used in embedded systems programming, and the AVR-GCC compiler is employed to compile the C code for the ATmega2560..

techno india university
WEST BENGAL

IIT Bombay eYantra
Engineering a better tomorrow

Schematic Diagram:



techno india university
WEST BENGAL

IIT Bombay e-yantra
Engineering a better tomorrow

Code:

```
/*
 * LED button.c
 *
 * Created: 14-01-2024 18:24:27
 * Author : souvik pal
 */

#define F_CPU 16000000
#include <avr/io.h>
#include <avr/interrupt.h>
#include <util/delay.h>


// Define debounce delay
#define DEBOUNCE_DELAY 50  // Adjust as needed

// Function to configure Interrupt switch
void interrupt_switch_config(void) {
	DDRE &= ~(1 << PE7);   // PORTE 7 pin set as input
	PORTE |= (1 << PE7);   // PORTE7 internal pull-up enabled
}

// Function to configure LED bar-graph displays
void LED_bargraph_config(void) {
	DDRJ = 0xFF;            // PORT J is configured as output for the first bar-graph
	PORTJ = 0x00;           // Output is set to 0 initially

	DDRK = 0xFF;            // PORT K is configured as output for the second bar-graph
	PORTK = 0x00;           // Output is set to 0 initially
}

// Function to Initialize PORTS
void port_init(void) {
	interrupt_switch_config();
	LED_bargraph_config();
}
```

```c
// Function to perform software debounce
uint8_t debounce_switch(void) {
	static uint8_t state = 0;  // Initial state
	uint8_t current_state = (PINE & (1 << PE7)) == 0;

	switch (state) {
		case 0:  // Initial or stable low state
		if (current_state == 0) {
			_delay_ms(DEBOUNCE_DELAY);
			state = 1;  // Move to the next state
		}
		break;

		case 1:  // Transition from low to high detected
		if (current_state == 1) {
			_delay_ms(DEBOUNCE_DELAY);
			state = 2;  // Move to the next state
			} else {
			state = 0;  // Reset to the initial state
		}
		break;

		case 2:  // Stable high state
		if (current_state == 1) {
			return 1;  // Debounced high
			} else {
			_delay_ms(DEBOUNCE_DELAY);
			state = 0;  // Move back to the initial state
		}
		break;
	}

	return 0;  // Default: debounced low
}

// Main Function
int main(void) {
	port_init();

	while (1) {
		if (debounce_switch()) {
			// Control Bar-graph 1 connected to PORTJ
			for (uint8_t i = 0; i < 8; i++) {
				PORTJ = (1 << i);  // Turn on the i-th LED
				_delay_ms(1);    // Adjust delay as needed
			}

			_delay_ms(1);  // Delay before turning off LEDs

			// Control Bar-graph 2 connected to PORTK
			for (uint8_t i = 0; i < 8; i++) {
				PORTK = (1 << i);  // Turn on the i-th LED
				_delay_ms(1);    // Adjust delay as needed
			}

			_delay_ms(1);  // Delay before starting the next sequence
		}
	}

	return 0;
}
```

techno india university
WEST BENGAL

IIT Bombay eYantra
Engineering a better tomorrow

Screenshots and Video link of simulation (Uploaded on Google drive):

https://drive.google.com/drive/folders/1o5h14FKvDqbCZC3hvR78rFTKH-swCFW1?usp=drive_link

References:

ATmega640/1280/1281/2560/2561 datasheet - Microchip Technology
https://ww1.microchip.com/downloads/en/devicedoc/atmel-2549-8-bit-avr-microcontroller-atmega640-1280-1281-2560-2561_datasheet.pdf

E-yantra Resources
https://elsi.e-yantra.org/resources

Microchip studio
https://www.microchip.com/en-us/tools-resources/develop/microchip-studio#Downloads

Proteus software
https://1drv.ms/u/s!Aq8DaZDYUmeDgVSFHtge3PjCsjGJ?e=fjB0o7

Smith, J. (2022). Toggle LED Bar-graphs with Interrupt Switch. [Microcontroller project]
https://homediyelectronics.com/projects/led_bar_graph_with_transistors/