Smart Home Energy Monitor

Rodney Tayebwa & Remi Pilon

Course: NET3001 Computer Systems

Professor: Mehdi Niknam

Due on April 10th, 2024

Table of Contents:

Cover page (1)

Table of contents (2)

Brief description of the system (3)

Circuit Diagram (4)

Components and topics used in the project (5)

Detailed explanation of the system (6)

Reflection (6)

Brief description of the system:

The Smart Home Energy Monitor is an embedded system designed to monitor and optimize energy usage within a home. It tracks electricity consumption in real-time and provides users with insights to help them make informed decisions about energy usage and conservation.

This project aims to empower homeowners to monitor and manage their energy usage more effectively, promoting energy conservation and cost savings while reducing environmental impact.

Circuit Diagram:

Components and topics used in the project:

Mandatory Topics and Components:

GPIO: Used for interfacing with energy monitoring sensors and controlling output devices.

Timers: Scheduled tasks for data logging and reporting.

Interrupts: Handle events such as sensor readings or user inputs.

LEDs: Indicate energy consumption levels or system status.

Push Buttons: Enable user interaction for viewing data or configuring settings.

7-segment Display: Display real-time energy consumption data.

LCD: Display detailed energy usage statistics and system status.

Shift Registers: Expand GPIO capabilities for driving multiple LEDs or displays efficiently.

Optional Topic and Components:

USART: Enable communication with external devices or data logging systems.

Active or Passive Buzzers: Provide audible alerts for high energy consumption levels or system warnings.

Potentiometer: Allow users to adjust display brightness or contrast.

Detailed explanation of the system:

// Initialize system components

usart\_init()

lcd\_init()

setup\_shift\_register\_pins()

button\_init()

setup\_LED\_pins()

timer1\_init()

// Enable global interrupts

enable\_interrupts()

// Main program loop

while (true) {

// Control LEDs based on button states

if (buttonState1) {

turn\_on\_LED\_1()

} else {

turn\_off\_LED\_1()

}

if (buttonState2) {

turn\_on\_LED\_2()

} else {

turn\_off\_LED\_2()

}

// Display button states on 7-segment display

display\_button\_states\_on\_7\_segment(buttonState1, buttonState2)

// Transmit received USART data to LCD

if (received\_char != '\0') {

display\_received\_data\_on\_LCD(received\_char)

clear\_received\_char()

}

}

Reflection: