**Part 1: Program Code**

/\*

\* Author: Written By Araam Zaremehrjardi

\* Title: Lab 7 - CPE 301 (10/24/2000)

\* Revision: 1

\* Description: A program to count data from a EEPROM and display on a 7 Segment Display

\*/

/\*

\* M\_EEPROM\_MEMORY\_ADDRESS is a macro defined to be used to

\* only store the "counter" variable data at the specific

\* memory address of 0x0001

\*/

#define M\_EEPROM\_MEMORY\_ADDRESS 0x0001

/\*

\* character\_array is used to decode values given from the variable "counter"

\* and turn on specifc LED lights for the 7 Segment Display on PORT C;

\*/

unsigned char character\_array[16] = {0x7F, 0x41, 0xB3, 0xB6, 0xD4, 0xE6, 0xE7, 0x34, 0xF7, 0xF4, 0xF5, 0xC7, 0x63, 0x97, 0xE3, 0xE1};

/\*

\* Intialize PORTS to their memory addresses in the microcontroller for later use.

\*/

volatile unsigned char\* DDR\_C = (unsigned char\*) 0x27;

volatile unsigned char\* PORT\_C = (unsigned char\*) 0x28;

volatile unsigned char\* DDR\_K = (unsigned char\*) 0x107;

volatile unsigned char\* PORT\_K = (unsigned char\*) 0x108;

volatile unsigned char\* PIN\_K = (unsigned char\*) 0x106;

volatile unsigned int\* EEPROM\_ADDR\_REG = (unsigned int\*) 0x41;

volatile unsigned char\* EEPROM\_DATA\_REG = (unsigned char\*) 0x40;

volatile unsigned char\* EEPROM\_CNTRL\_REG = (unsigned char\*) 0x3F;

/\*

\* setup function Iintializes ports to become input or outputs.

\* specifically PORT C is used for outputting binary "1" or "0"

\* to the 7 Segment Display. PORT K is used for operating input

\* that increments the variable "counter." During setup, counter

\* and the 7 Segment display is updated on intial boot up to display

\* data last saved to the EEPROM.

\*/

void setup(void) {

Serial.begin(9600);

\*DDR\_C = 0xF7;

\*DDR\_K |= 0xFE;

\*PORT\_K |= 0x01;

unsigned int counter = 0;

counter = eeprom\_read(M\_EEPROM\_MEMORY\_ADDRESS);

Serial.println(counter);

display\_number(counter);

}

/\*

\* The loop function with two debouncing techniques is used to read

\* input from the push button implemented on PORT K. This button

\* on push increments the "counter" variable which then is updated on

\* the EEPROM.

\*/

void loop(void) {

unsigned int counter = eeprom\_read(M\_EEPROM\_MEMORY\_ADDRESS);

if (!(\*PIN\_K & 0x01)) {

for (volatile unsigned int index = 0; index < 1300; index++);

if (!(\*PIN\_K & 0x01)) {

Serial.println("increment");

counter++;

counter %= 16;

Serial.println(counter);

eeprom\_save(M\_EEPROM\_MEMORY\_ADDRESS, counter);

while (!(\*PIN\_K & 0x01));

}

}

display\_number(counter);

}

void display\_number(unsigned char number) {

\*PORT\_C = character\_array[number];

}

void eeprom\_save(unsigned int address, unsigned char data\_input) {

while(\*EEPROM\_CNTRL\_REG & 0x02);

\*EEPROM\_ADDR\_REG = address;

\*EEPROM\_DATA\_REG = data\_input;

\*EEPROM\_CNTRL\_REG |= 0x04;

\*EEPROM\_CNTRL\_REG &= ~(0x02);

\*EEPROM\_CNTRL\_REG |= 0x02;

}

unsigned char eeprom\_read(unsigned int address) {

while(\*EEPROM\_CNTRL\_REG & 0x02);

\*EEPROM\_ADDR\_REG = address;

\*EEPROM\_CNTRL\_REG |= 0x01;

return \*EEPROM\_DATA\_REG;

}

**Part 2: Circuit Design**

A picture containing indoor, computer, person, computer

Description automatically generated