Economy Priced Industrial Gas Sensor

Anh Doan   
*College of Science Cal State East Bay*  
Hayward CA, USAadoan2@horizon.csueastbay.eduYash Naik  
*College of Science*  
*Cal State East Bay*Hayward CA, USA  
ynaik@horizon.csueastbay.edu  
Eric Ramos *College of Science*  
*Cal State East Bay*HaywardCA,USA eramos@horizon.csueastbay.edu

# Introduction

A do-it-yourself board has been a dream for many of us due to complex parts and the software to run the board. There are several solutions in the market in the form of Raspberry Pi, Arduino, or a Beaglebone board. Sure, the above-mentioned boards are very neat in their own way but, they still are not a true do-it-yourself. The design files are not available and difficult hardware manufacturing process prohibits a new comer from designing a board similar to those off-the-shelf products. Off-the-shelf Linux boards help to skip certain steps in development phase but create new problems in exchange. Take the Raspberry Pi, if you wanted to increase the memory capacity or add another peripheral, well that is out of the question as all the parts are solder on without any space for addition. Finally, modules and boards always have shorter lifetime than the components they are made of. Off-the-shelf boards will become obsolete faster than certain types of CPUs, memory chips, peripherals etc. so designing our own custom board specifically targeting our own requirements will inevitably give more efficiency and better control in production.

# Block Diagram

A screenshot of a social media post

Description automatically generated

This design is centered on an ATMEGA1284P 8- bit processor made by AVR. The processor has 256 kb of on-board flash for user code. The design utilized a linear regulator with a resistor divider Feedback loop to ensure the processor receives no transient voltage during its operation. JTAG pins are exposed in order to debug and monitor system operation.

# Schematics

Revision A

A close up of a map

Description automatically generated

Three primary devices on the schematic are the processor, two MQ135 sensor, and a 16x2 LCD.

# layouts

Revision B

A circuit board

Description automatically generated

Our first Printed Circuit Board was complicated since we did not have experience with the software. However, we were able to manage to learn the basics of the software and created our first layout in eagle. The first mistakes that we made in the layout was that we miscalculated the dimension of the LCD, and that caused us to put one of the sensors on the back of the board.

# (software section)

A close up of text on a white background

Description automatically generated

#define F\_CPU 16000000UL

#include <avr/io.h>

#include "lcd.h"

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <math.h>

#include <util/delay.h>

void adc\_init()

{

ADMUX = (1<<REFS0);

ADCSRA = (1<<ADEN)|(1<<ADPS2)|(1<<ADPS1)|(1<<ADPS0);

}

uint16\_t adc\_read(uint8\_t ch)

{

//select adc channel 0-7

//AND with 111 will keep value of ch between 0 and 7

ch &=0b00000111;

ADMUX = (ADMUX & 0xF8) | ch;

//start single conversion by writing "1" to ADSC

ADCSRA |= (1<<ADSC);

//wait for conversion to complete

while(ADCSRA & (1<<ADSC));

return(ADC);

}

int main(void)

{

char buffer[20];

uint16\_t sensor1;

uint16\_t sensor2;

MCUCR = (1<<JTD); //disable JTAG on PORTC

MCUCR = (1<<JTD);

lcd\_init(LCD\_DISP\_ON\_CURSOR);

\_delay\_ms(100);

lcd\_clrscr();

\_delay\_ms(2);

DDRD = 0b11111111; // connect LED to PD0

adc\_init();

\_delay\_ms(50);

while (1)

{

lcd\_clrscr();

lcd\_puts("CO2");

lcd\_gotoxy(0,1);

sensor1 = adc\_read(0);

sensor2 = adc\_read(1);

sensor1 = sensor1 \* 10;

sensor2 = sensor2 \* 4;

if (sensor1 - sensor2 < 75)

{

itoa(sensor1,buffer,10);

lcd\_puts(buffer);

PORTD = 0b00000000;

}

else

{

lcd\_puts("sensor error");

PORTD = 0b00000010;

}

\_delay\_ms(100);

}

}



# (milestone)

We achieved all 5 milestones of the project. We successfully soldered all the component on the PCB and achieved power up. Next, we flashed a quick blink led program to test all the GPIOs. Finally, we flashed the gas sensor program and we achieved a functioning system

# (reach goal)

Our reach goal would be to solder on a 5v relay in which we can control another appliance. Time did not permit us to incorporate this feature onto our board.

# (conclusion)

We were able to bring the board from conception to completion in a total of 2 weeks. The board total cost is 30 USD, to include part and manufacturing and shipping. We had a schematic sent out to the manufacturer in only 2 days. We have made extreme progress in debugging the board using standard lab equipment like the oscilloscope and the multi-meter. We were fortunate that the programmer for the MCU was not expensive at all. The MCU had plenty of on-board flash for us to include as many libraries as we needed. Although we started our senior design project with an objective to design a multi-purpose board that could run Linux. However, due to crunch of time and lack of experience, we were unable to program our first board. This led us to incarnate our board with a less complex MCU. With board two, we were able to achieve all the milestones for the class and meet our project objectives. This project taught us the breakneck pace of rapid prototyping and it will serve as a good lesson for us in future projects.