IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Design And Implementation Of A Data Acquisition System Using LPC1768 Microcontroller

Dr.P.Thimmaiah[*1], Dr.B.Ashraf ahamed*[2], S. Khaja Hussian[*3] and M Vishnu Vardhan[*4]

*1 Asst.Professor, Department of Physics & Electronics, Sri Krishnadevaraya University, Anantapur, A.P, India-515003

*[2] Senior Technical trainer, Indian Institute of Embedded Systems, Bangalore, Karnataka (State) - 560095

*[3&4] Research Scholar, Department of Physics & Electronics, Sri Krishnadevaraya University, Anantapur, A.P, India-515003

Abstract: Data Acquisition Systems (DAS) are essential in areas such as industrial automation, environmental monitoring, and instrumentation. This paper describes the design and implementation of a low-cost, real-time data acquisition system based on the ARM Cortex-M3 LPC1768 microcontroller. The proposed system can collect data from multiple analog input channels, process the acquired signals, and transmit the information to a computer or an IoT server through USB or serial communication. The system utilizes the built-in peripherals of the LPC1768, including Analog-to-Digital Converters (ADCs), timers, and UART interfaces, to ensure efficient data acquisition and communication. The developed system offers a reliable, flexible, and cost-effective solution for real-time monitoring applications.

Keywords: LPC 1768 (CORTEX-M3), Embedded systems, Data Acquisition, 16X2 LCD, sensors

1. INTRODUCTION

Data acquisition refers to the process of collecting signals that represent real-world physical parameters and converting them into digital data that can be processed and analyzed by a computer. This process is widely used in monitoring, measurement, and control systems. With the advancement of embedded technology, modern microcontrollers provide integrated peripherals that simplify the development of efficient data acquisition systems.

The LPC1768 microcontroller, based on the ARM Cortex-M3 core, includes several built-in features such as 12-bit Analog-to-Digital Converters (ADCs), multiple serial communication interfaces, timers, and general-purpose input/output (GPIO) pins. These capabilities make it well suited for embedded data acquisition system (DAS) applications.

This paper presents the design, architecture, implementation, and testing of a data acquisition system developed using the LPC1768 microcontroller.

2. REQUIREMENT TO THE DATA ACQUISITION SYSTEM

The main component use in the Data Acquisition System is the ADC unit connected to the POT, temperature sensor, pressure sensor and humidity sensor, Gas sensor. The ADC is used to take analog inputs from sensors and is concerted into digital values. As we know the ADC in LPC1768 is of 12-bit resolution it will give a digital value of range 0 to 4095.we need to convert these digital values in sensors readings like temperature, volts etc., this is done by using some formulas. These values are continuously display on LCD. Based on the sensor values the total Data Acquisition System works, the controlling of these sensor values when it exceeds the maximum limit is done by motors through relays.

3. SYSTEM ARCHITECTURE

3.1 LPC1768 Features

- 12-bit ADC with up to 8 channels
- 512 KB Flash and 64 KB SRAM
- UART, SPI, I2C, CAN, and USB
- 100 MHz clock speed
- Cortex-M3 processor core

3.2 Hardware Components

- LPC1768 microcontroller
- Analog sensors (e.g., temperature, light, voltage sensors)
- Serial-to-USB converter (e.g., FT232)
- Display (LCD or OLED)
- PC/laptop

3.3 Software Tools

- Keil uVision (IDE)
- CMSIS libraries for ARM
- RealTerm or serial terminal for data display
- Optional: Python/Node.js script for plotting/logging

4. SYSTEM BLOCK DIAGRAM

4.1 Analog Signal Acquisition

The LPC1768's ADC supports 12-bit resolution and a sampling rate of up to 200 kSPS. Analog signals from sensors are connected to the ADC input pins. The ADC is configured for burst mode to allow continuous data sampling. The LPC1768 development board has internally connected the Temperature and POT on board as shown in figure 2.

4.2 Data Processing

A simple digital filtering algorithm (e.g., moving average) is applied to smooth out sensor readings. The data is then formatted and transmitted to LCD.

4.3 Communication

UART communication is configured at 9600 bps. Data is sent to the PC in a structured format (e.g., JSON or CSV). Optional USB or CAN interfaces can be used for higher-speed communication or industrial integration.

4.4 Power Management

The system operates at 3.3V, with on board regulators ensuring stable voltage for ADC accuracy. The low-power modes of the LPC1768 are optionally used to reduce power consumption in battery-powered applications.

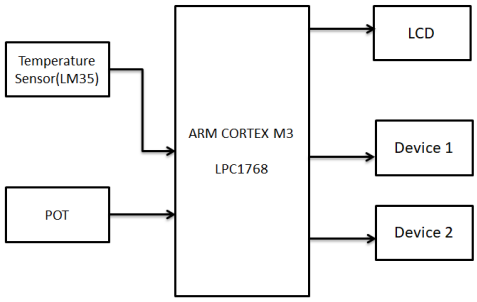

Figure 1: Block Diagram of Data Acquisition System

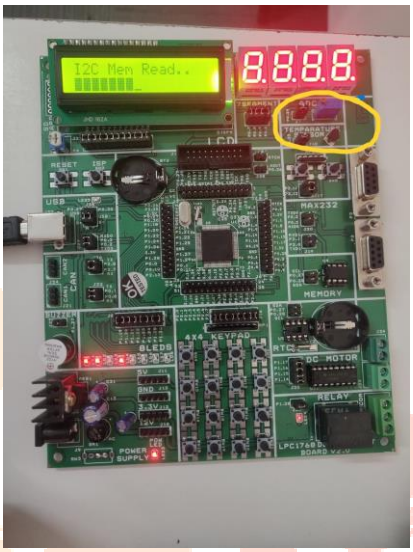

Figure 2. Temperature and POT connection

5. DESIGN AND IMPLEMENTATION

In this study controlling the Physical parameter temperature and to measure the temperature value using an LM35 sensor which measures the temperature continuously. In ARM there are inbuilt ADC and DAC. The ADC to convert the analog data from the sensor and convert it to the digital value and the microcontroller continuously reads the temperature from the sensor and will be compared with the set point. Devices 1 and 2 are the two devices which are going to control these devices may be the boilers (or) machines. The measured temperature from the measured sensor will be compared continuously with the set point if the measured temperature is below the set point the ARM CORTEX-M3 microcontroller will switch on the heater and if the temperature is above the set point it will switch on the heater

5.1 LM35 Sensor

There are so many kinds of sensors. Sensors Applications covers all major fields of applications. In this project to control temperature (physical parameters), for this purpose LM35 temperature sensor used to measure the temperature. The LM35 thus has an advantage over linear temperature sensors calibrated in ° Kelvin, as the user is not required to subtract a large constant voltage from its output to obtain convenient Centigrade scaling. The LM35 does not require any external calibration or trimming to provide typical accuracies of ±1⁄4°C at room temperature and ±3⁄4°C over a full −55 to +150°C temperature range. Low cost is assured by trimming and calibration at the wafer level. The LM35's low output impedance, linear output, and precise inherent calibration make interfacing to readout or control circuitry especially easy. It can be used with single point power supplies, or with plus and minus supplies. As it draws only 60 µA from its supply, it has very low self-heating, less

than 0.1°C in still air. The LM35 is rated to operate over a −55° to +150°C temperature range, while the LM35C is rated for a −40° to +110°C range (−10° with improved accuracy).

- Calibrated directly in ° Celsius (Centigrade)
- Linear + 10.0 mV/°C scale factor
- 0.5°C accuracy guarantee able (at +25°C)
- Rated for full −55° to +150°C range
- Suitable for remote applications
- Low cost due to wafer-level trimming
- Operates from 4 to 30 volts
- Less than 60 µA current drain
- Low self-heating, 0.08°C in still air
- Nonlinearity only ±1⁄4°C typical
- Low impedance output, 0.1 W for 1 mA load

5.2 ADC CONFIGURATION

An analog-to-digital converter (abbreviated ADC, A/D or A to D) is a device that converts a continuous physical quantity (usually voltage) to a digital number that represents the quantity's amplitude. The conversion involves quantization of the input, so it necessarily introduces a small amount of error. The inverse operation is performed by a digital-to-analog converter (DAC). Instead of doing a single conversion, an ADC often performs the conversions ("samples" the input) periodically. The result is a sequence of digital values that have converted a continuous-time and continuous- amplitude analog signal to a discrete-time and discrete- amplitude digital signal. An ADC may also provide an isolated measurement such as an electronic device that converts an input analog voltage or current to a digital number proportional to the magnitude of the voltage or current. However, some non- electronic or only partially electronic devices, such as rotary encoders, can also be considered ADCs.

Figure 3: Analog to Digital converter

Pins relating to ADC Module of LPC1768/LPC1769

Adc Channel	Port Pin	Pin Functions	Associated PINSEL Register
AD0	P0.23	0-GPIO, 1-AD0[0], 2-I2SRX_CLK, 3-CAP3[0]	14,15 bits of PINSEL1
AD1	P0.24	0-GPIO, 1-AD0[1], 2-I2SRX_WS, 3-CAP3[1]	16,17 bits of PINSEL1
AD2	P0.25	0-GPIO, 1-AD0[2], 2-I2SRX_SDA, 3-TXD3	18,19 bits of PINSEL1
AD3	P0.26	0-GPIO, 1-AD0[3], 2-AOUT, 3-RXD3	20,21 bits of PINSEL1
AD4	P1.30	0-GPIO, 1-VBUS, 2- , 3-AD0[4]	28,29 bits of PINSEL3
AD5	P1.31	0-GPIO, 1-SCK1, 2- , 3-AD0[5]	30,31 bits of PINSEL3
AD6	P0.3	0-GPIO, 1-RXD0, 2-AD0[6], 3-	6,7 bits of PINSEL0
AD7	P0.2	0-GPIO, 1-TXD0, 2-AD0[7], 3-	4,5 bits of PINSEL0

5.3 LPC176x ADC Registers

Register	Description
ADCR	A/D Control Register: Used for configuring the ADC
ADGDR	A/D Global Register: This register contains the ADC's DONE bit and the result of the most recent A/D Conversion
ADINTEN	A/D Interrupt Enable Register
ADDR0-ADDR7	A/D Channel Data Register: Contains the recent ADC values for respective channel
ADSTAT	A/D Status Register: Contains DONE & OVERRUN flag for all the ADC channels

ADCR: A/D Control Register

ADCR								
31:28	27	26:24	23:22	21	20:17	16	15:8	7:0
Reserved	EDGE	START	Reserved	PDN	Reserved	BURST	CLCKDIV	SEL

Bit 7:0 – SEL : Channel Select These bits are used to select a particular channel for ADC conversion. One bit is allotted for each channel. Setting the Bit-0 will make the ADC to sample AD0[0] for conversion. Similarly setting bit-7 will do the conversion for AD0[7].

Bit 15:8 – CLCKDIV : Clock Divisor The APB clock (PCLK_ADC0) is divided by (this value plus one) to produce the clock for the A/D converter, which should be less than or equal to 13 MHz.

Bit 16 – BURST This bit is used for BURST conversion. If this bit is set the ADC module will do the conversion for all the channels that are selected(SET) in SEL bits. CLearing this bit will disable the BURST conversion.
Bit 21 – PDN : Power Down Mode Setting this bit brings ADC out of power down mode and makes it operational. Clearing this bit will power down the ADC.

Bit 24:26 – START When the BURST bit is 0, these bits control whether and when an A/D conversion is started:
000 - Conversion Stopped
001-Start Conversion Now
The remaining cases (010 to 111) are about starting conversion on occurrence of edge on a particular CAP or MAT pin.

Bit 27 – EDGE This bit is significant only when the START field contains 010-111. It starts conversion on selected CAP or MAT input.
0 – On Falling Edge
1 - On Rising Edge

ADGDR: A/D Global Data Register

ADGDR					
31	27	26:24	23:16	15:4	3:0
DONE	OVERRUN	CHN	Reserved	RESULT	Reserved

Bit 15:4 – RESULT This field contains the 12bit A/D conversion value for the selected channel in **ADCR.SEL** The vale for this register should be read once the conversion is completed ie DONE bit is set.

Bit 26:24 - CHN : Channel These bits contain the channel number for which the A/D conversion is done and the converted value is available in RESULT bits(e.g. 000 identifies channel 0, 011 channel 3...).

Bit 27 – OVERRUN This bit is set during the BURST mode where the previous conversion data is overwritten by the new A/D conversion value.

Bit 31 – DONE This bit is set to 1 when an A/D conversion completes. It is cleared when this register is read and when the ADCR is written. If the ADCR is written while a conversion is still in progress, this bit is set and a new conversion is started.

6. STEPS FOR CONFIGURING ADC

1. Configure the GPIO pin for ADC function using PINSEL register.
2. Enable the CLock to ADC module.
3. Deselect all the channels and Power on the internal ADC module by setting ADCR.PDN bit.
4. Select the Particular channel for A/D conversion by setting the corresponding bits in ADCR.SEL
5. Set the ADCR.START bit for starting the A/D conversion for selected channel.
6. Wait for the conversion to complete, ADGR.DONE bit will be set once conversion is over.
7. Read the 12-bit A/D value from ADGR.RESULT.

7. PROGRAM

```
#include<LPC17xx.h>
#include "stdutils.h"
#include<stdio.h>
#define RCLR LPC_GPIO0->FIOCLR
#define RSET LPC_GPIO0->FIOSET
#define Rselect 1<<10
#define SBIT_BURST      16u
#define SBIT_START      24u
#define SBIT_PDN        21u
#define SBIT_EDGE       27u
#define SBIT_RESULT     4u
#define SBIT_CLCKDIV    8u

const int numreadings = 25;

void LCD_INIT();
void CMD(unsigned char x);
void display(char x);
void string(char displaystring[]);
void delay();
void delay2(int a);
```

```
int main(){
        uint32_t adc_result;
        char sval[20];
        SystemInit();
        LPC_SC->PCONP |= (1 << 12);
        LPC_ADC->ADCR = ((1<<SBIT_PDN) |
          (5<<SBIT_CLCKDIV));
        LPC_PINCON->PINSEL1|= 0x01<<18;
        LPC_GPIO0->FIODIR|=1<<11|1<<10|0xFF<<15;
        LCD_INIT();
        string("This is LCD");
        LPC_ADC->ADCR  |= 1<<2; //0x03;
        while(1){
        CMD(0x01);
        delay2(1);
        LPC_ADC->ADCR |= 1<<24;
        while((LPC_ADC->ADGDR &(0x01<<31))==0);
        adc_result = (LPC_ADC->ADGDR >> SBIT_RESULT);
            adc_result = adc_result & 0xfff;
        float volt =(3.3/4095);
```

```
void string_write(char *str);

int currentreadings = 0;
float readings[numreadings];
float sum = 0; float avarage = 0;
```

```
        volt = (volt *  adc_result) * 100;
        readings[currentreadings] = volt;
        sum += volt;
        currentreadings++;
        if(currentreadings>= numreadings) {
                avarage = sum/numreadings;
                sprintf(sval, "%.2f Deg", avarage);
                delay2(1);
                string_write(sval);
                sum = 0;
                currentreadings = 0;
                }
        }

}
```

```
void LCD_INIT()
{
        CMD(0x38);
        CMD(0x01);
        CMD(0x0C);
        CMD(0x0F);
        CMD(0x80);
}

void CMD(unsigned char x)
{
        RCLR |=Rselect;
        LPC_GPIO0-
>FIOCLR|=0xFF<<15;
        LPC_GPIO0->FIOSET|=x<<15;
        LPC_GPIO0->FIOSET|=1<<11;
        delay();
        LPC_GPIO0->FIOCLR|=1<<11;
}

void display(char x)
{
        RSET |=Rselect;
        LPC_GPIO0-
>FIOCLR|=0xFF<<15;
        LPC_GPIO0->FIOSET|=x<<15;
        LPC_GPIO0->FIOSET|=1<<11;
        delay();
        LPC_GPIO0->FIOCLR|=1<<11;
}
```

```
void string(char displaystring[])
{
        int s=0;
        while(displaystring[s])
{
                display(displaystring[s++]);
        }
}
void delay()
{
        int i;
        for(i=0;i<500000;i++);
}
void delay2(int a)
{
        int i,j;
        for(i=0;i<a;i++)
        {
                for(j = a; j<10000; j++);
        }
}
void string_write(char *str)
{
        while(*str != '\0')
{
                display(*str);
                str++;
        }
}
```

8. RESULTS AND DISCUSSION

The DAS was tested with a temperature and a light sensor. The analog voltage outputs were accurately digitized by the LPC1768 and transmitted to a LCD at regular intervals. The system demonstrated:

- 0.5°C accuracy in temperature measurement
- Stable real-time data acquisition at 1 kHz sampling rate
- Reliable communication over UART up to 115200 bps

9. APPLICATIONS

- Environmental monitoring systems
- Laboratory instrumentation
- Industrial automation
- Internet of Things (IoT) sensor nodes

10. CONCLUSION

This paper presents the development of a practical and efficient data acquisition system using the LPC1768 microcontroller. The proposed system operates in two modes: data acquisition mode and data actuation mode. In both modes, the compiled program (.bin file) is uploaded to the LPC1768 development board through a USB interface.

During the data acquisition mode, the Analog-to-Digital Converter (ADC) module of the LPC1768 is tested and used to acquire input signals. The processed results are displayed on an LCD and based on the acquired data, devices 1 and 2 are controlled accordingly. The implemented system demonstrates the capability of the LPC1768 microcontroller to perform reliable data acquisition and control operations in embedded applications.

11. REFERENCES

[1] NXP Semiconductors, "LPC1768 Datasheet," www.nxp.com
[2] ARM, "Cortex-M3 Technical Reference Manual," ARM Holdings.
[3] Keil MDK ARM Development Tools, https://www.keil.com
[4] Umar Hamid, Rahim Ali Qamar, Mohsin Shahzad,"2013," PC Based Data Acquisition and Signal Processing for Underwater Sensor Arrays"; Proceedings of 2013 10th International Bhurban Conference on Applied Sciences & Technology (IBCAST).
[5] Alen Rajan and Aby K.Thomas; "ARM Based Embedded Web Server for Industrial Applications," International Conference on Computing and Control Engineering, April2012.
[6] Manivannan, M and Kumaresan, N; "Design of On-line Interactive Data Acquisition and Control System for Embedded Real Time Applications," International Conference on Emerging trends in Computer and Information Technology, pp.551-556, March 2011.
[7] Wang Jiannong, Wang Wei, ICEMI"2011," The Common Data Acquisition System Based on Arm9"; The Tenth International Conference on Electronic Measurement & Instruments.
[8] Junhua Yang; Zhien Shang and Tao XinG, "Intelligence Monitoring System Based on ARM and Information Fusion," International Conference on Electric Information and Control Engineering, pp.487- 490, April 2011.
[9] Karia, D.C., Adajania, V., Agarwal, M and Dandekar, S.; "Embedded Web Server Application Based Automation and Monitoring System," International Conference on Signal Processing, Communixation, Computing and Networking Technologies, pp.634-637, July 2011.
[10] Wu Xiguang, Li Bonian, Zhao Likai and Zhang Minghu; "An embedded real-time remote monitoring system based on B/S mode," international Conference on Mechatronic Science, Electrical Engineering and Computer, pp.2135-2138, Aug 2011
[11] Yujun Bao, Xiaoyan Jiang, ICCASM "2010," Design of USB Data Acquisition Card which based upon ARM kernel Microprocessor";2010 International conference on computer Application and System modelling.
[12] LPC1768 User Manual" Philips Semiconductors.
[13] www.mbed.org
[14] http://en.wikipedia.org/wiki/Universal serial bus.
[15] Zhao Ruimei and Wang Mei; "Design of ARM-based Embedded Ethernet Interface," 2nd International Conference on Computer Engineering and technology, pp.V4-268-V4- 270, April 2010.

[16] Rui Yang, Hong Cai and Ming zhang; "Research and Implement of Ethernet Interface Based on Embedded System," Second International Symposium on Computational Intelligence and Design, pp.288-291, Dec 2009.
[17] Zhan mei-qiong and Ji chang-peng; "Research and Implementation of Embedded Web Server," International Conference on Multimedia and Information Technology, pp.123-125, Dec 2008.
[18] Wu Min-hua; "Research for the Embedded Web Server," Microwave Conference, pp.776-779, Sept 2008.
[19] Masato Shimano, Futoshi Okazaki, Yoshihiro Saito, Akiya Fukui, Takako Nonaka and Tomohiro Hase; "Small, Embedded Web Server for Home Appliances with Embedded MPU and Real-time Operating System," pp.1-3, June 2007.