

Microcontroller Based System for Control and Monitoring

Preyas.T. Shah, Ashish.R.Patel

Abstract— This paper reviews the design of the system that can be used for industrial control and monitoring. This system is equipped with the necessary hardware so as analog and digital inputs and outputs can be interfaced with it. Digital and analog inputs and outputs are as per industry standard. For ease and flexibility for employing this system, the software interface is provided by programming various functional blocks in the system.

Index Terms— PLC, DCS, CTR.

I. INTRODUCTION

In Process Instrumentation/Batch Applications there are different systems available for monitoring and control in industry. Typical Systems used in industry are PLC (Programmable Logic Controllers) and DCS (Distributed Control Systems). These systems are very complex and costly, so they are not available for small scale industries where control is through relay based system.

So, a system is proposed here that gives us a cost effective solution for small scale industrial automation. This system employs digital and analog interfacing circuitry in one controller giving a cost effective solution.

Digital inputs typically are pressure switches, level switches, liquid level switches, limit switches etc. Analog input devices are potentiometer, Thermocouples, Temperature sensors, flow sensors, optical and magnetic sensors etc. Digital or on/off outputs are solenoids, solenoid actuator, relays and indicators. Analog output devices are 4-20mA indicators, stepper motor, servo motor etc.

Digital inputs operate at 12v or 24v. Analog inputs and outputs are standard 4-20mA. Analog and digital inputs and outputs interfacing must be provided in the system.

For using this system in industrial applications, the software interface should be provided in the system. Various functional blocks are required for the logic design, mathematical operations on the data that is obtained through sensors, controlling the operations of various devices, Device driving mechanisms etc. These functional blocks are enlisted below:

Digital logic blocks:

AND Logic

OR Logic

NOT Logic

Device Drive Logics

Analog functional Blocks:

Manuscript received September 10, 2013.

Preyas. T. Shah is a student of Electronics Engineering in Maharaja Sayajirao University of Baroda.

Ashish. R. Patel is a student of Electronics Engineering in Maharaja Sayajirao University of Baroda.

High/Low Select
High/Low Compare
Square Root
Analog Transfer
PID Control
Timers

These functional blocks provide the interface between hardware and industrial applications. When this system is used for industrial control, analog and digital inputs and outputs that are required need to be interfaced with system and using functional blocks described earlier, the logic for industrial control is designed and implemented.

II. HARDWARE

A. Block Diagram

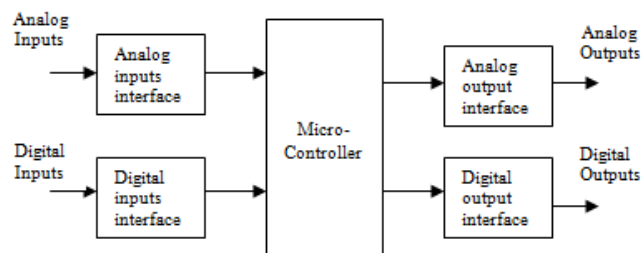


Figure 1: Block Diagram of the system

Figure 1 shows the basic block diagram of the system. In hardware, analog and digital inputs and outputs are interfaced to microcontroller via interface as shown in figure.

B. Analog Input Interface

Analog input devices are potentiometer, Thermocouples, Temperature sensors, flow sensors, optical and magnetic sensors etc. Analog inputs used in industry are typically 4-20mA. These inputs can be interfaced to the microcontroller through ADC (If microcontroller has inbuilt ADC, as in many case, the inputs can be given directly to the microcontroller port). ADC input is analog voltage signal, while 4-20mA signal is analog current signal, so this 4-20mA signal can be converted to 1-5 volts using 250ohm resistor. Normally, 250 Ω resistor is not available as a standard resistor, so 220 Ω or 270 Ω resistor can be used.

C. Digital Input Interface

Digital inputs typically are pressure switches, level switches, liquid level switches, limit switches etc. These digital inputs normally operate at 12V or 24V, whereas microcontroller supply voltage is 5V. Also, if any fault occurs due to short or open circuit, microcontroller can be damaged permanently. To prevent this to occur, digital inputs are normally given to

microcontroller after providing isolation using opto-isolators. These opto-isolators are available in IC, each IC contains two, four or eight opto-isolators.

Figure-2 shows typical digital input interfacing circuit. A diode is used for the reverse voltage protection because reverse breakdown voltage of opto-isolators is very low, this will clamp reverse voltage of opto-isolator LED to 0.7V. The values of R_1 and R_2 should be chosen according to the forward current rating of Opto-isolator, CTR (Current Transfer Ratio) of Opto-isolator. The value of R_2 should be chosen such that when current passes through LED, the output phototransistor is in saturation.

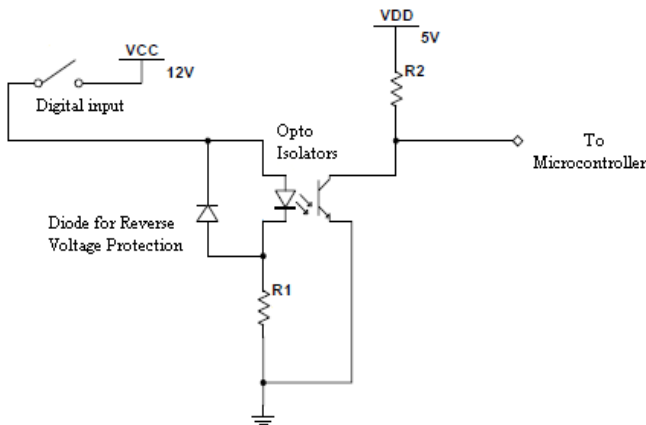


Figure 2: Digital Input Interfacing

D. Analog Output Interface

Analog output devices are 4-20mA indicators, stepper motor, servo motors. There must be suitable driver available for driving stepper motor and servo motors. 4-20mA Analog outputs are standard in industry. The output of the microcontroller port is given to the DAC (Digital to Analog Converter). If the analog output of DAC is voltage, then it has to be converted to current by using op-amp as voltage to current converter. Suitable Op-amp should be selected in order to avoid any loading problem. For more details on Op-amp and its applications, use reference [3].

E. Digital Output Interface

Digital outputs are solenoids, solenoid actuator, relays and indicators. Microcontroller ports provide low source or sink current (For more information, refer to microcontroller's datasheet). This current is not enough for driving digital outputs like Solenoid, Relays and Indicators etc. So, Suitable driver needs to be implemented for driving these devices. Depending upon the current requirement, suitable driver IC is selected.

So, all these interfacing circuitry has to be designed. This depends upon the number of analog and digital inputs and outputs selected for system implementation. Also, Microcontroller may not have sufficient number of ports available. In that case, Port Expander IC is used. Serial Port expanders are more used these days. SPI/I²C Serial Protocols can be used for serial port expanders.

III. SOFTWARE

A. Flow Chart

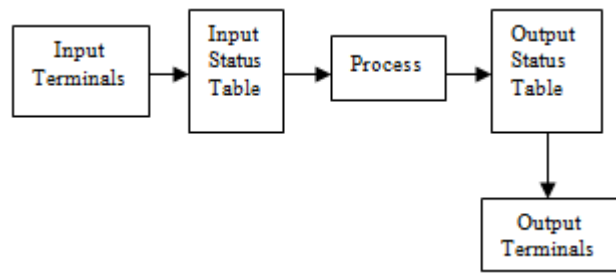


Figure 3: Flow Chart

Flow chart of the software is shown in figure 3. First of all, digital and analog inputs are scanned from the terminals; values of inputs are stored in input status table. Then process block is executed. Process cycle is the actual processing as per the requirement. It calls different function blocks, processes the inputs from input status table and output status table is updated accordingly. Then data is sent to output terminals from output status table. Various digital and analog outputs are connected to the output terminals.

Having understood the basic flow chart, now description of various functional blocks is given next.

IV. FUNCTION BLOCKS

ANALOG

A. PID

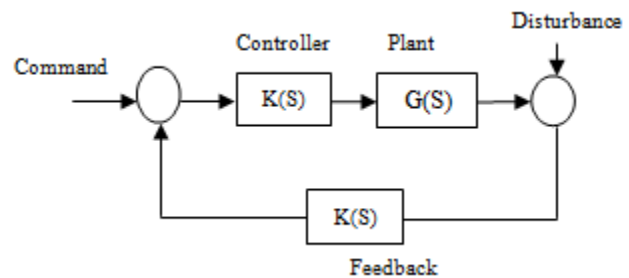


Figure 4: Basic Control System Block Diagram

Figure-4 Shows the basic block diagram of control system. The input to the system is the set-point. A feedback is given back to the system and compared with set point. Set Point is the desired value of the output. The difference between set point and feedback is given to the controller input which can be PID Controller. The output of the PID Block is used to drive the actuator. The Operation of the PID control block is dependent upon the values of parameters, K_p , K_i , K_d and set point. These parameters need to be tuned for a particular system for the proper operation of this algorithm.

Quantities like flow, temperature and voltage are not discrete signals but continuous ones. However, digital computers are used to manipulate sampled data values. So the digital PID control algorithm is implemented. More details on PID algorithms, practical aspects and tuning methods are given in reference [2].

The algorithm for PID Control is as shown below which can be implemented on digital computer.

1. Take error as input to PID Algorithm.
2. Calculate Proportional Term: $K_p * \text{error}$

3. Accumulate error
4. Limit the value of Error to avoid integral wind-up
5. Calculate Integration Term: $K_i * \text{Accumulated error}$
6. Calculate Proportional Term: $K_d * (\text{error} - \text{prevError})$
7. Add Proportional, Integration and Derivative Terms

The algorithm is modified if proportional and derivative kick needs to be removed. (More details in reference [2]).

B. AUTO MANUAL STATION

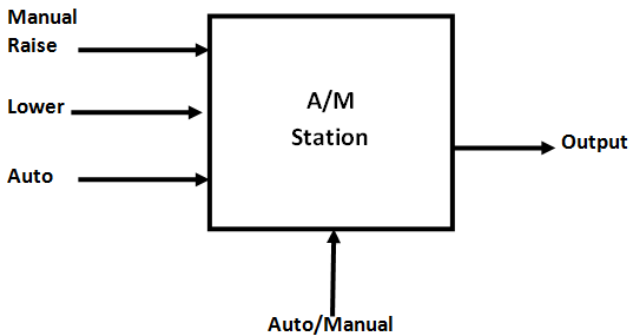


Figure 5: Auto/Manual Station

Auto manual station is used with PID Algorithm block. Auto/Manual select signal will decide whether the program is running in the automatic mode or manual mode. In manual mode the signals at the raise will cause the output to rise by some value, whereas the signal at the lower will cause the output to go down by some value. If the Program is in the automatic mode then the output of the PID is directly given as the output. It is useful to control the process manually if there is failure.

C. FUNCTION GENERATOR

Function generator is used to approximate nonlinear relationship between input and output. The input range is divided into four sections and linear output to input relationship is set for each of the four sections. Function generator block computes analog output related to input according to the linear relationship of four sections. Function generator can be used in shaping algorithm. It can be used as noise filter. It is used to convert analog input (4mA to 20mA) into physical data such as temperature or pressure for simplification of further processing. This block is used to realize the nonlinear response of sensors and actuators.

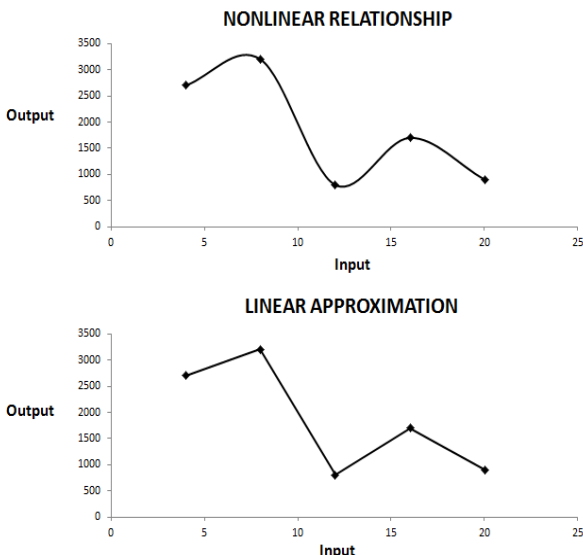


Figure 6: How non-linear characteristics can be linearly approximated by using function generator

Algorithm for Function Generator can be shown as follows:

Let, (x_1, y_1) , (x_2, y_2) , (x_3, y_3) and (x_4, y_4) are 4 points on graph of piecewise linear approximation of the characteristics. Let, x be the input to the function generator block.

```

if(x < x1) then y = x * y1 / x1; // First Region
else if(x < x2) then y = y1 + (x - x1) * (y2 - y1) / (x2 - x1);
//Second Region
else if(x < x3) then y = y2 + (x - x2) * (y3 - y2) / (x3 - x2);
//Third Region
else if(x < x4) then y = y3 + (x - x3) * (y4 - y3) / (x4 - x3);
//Fourth Region
else y = y4 //Does not fall within any region. Limit the value of
output.
    
```

D. HIGH SELECT

This functional block has 3 analog inputs (analog input1, analog input2 and analog input3) and one analog output. Analog output will be the highest of the three analog inputs.

E. LOW SELECT

This functional block has 3 analog inputs (analog input1, analog input2 and analog input3) and one analog output. Analog output will be the lowest of the three analog inputs.

F. HIGH LOW COMPARE

This functional block has one analog input and two digital outputs (high output and low output). If input analog signal has value greater than high limit, high output will be high and low output will be low. If analog signal has value less than low limit then high output will be low and low output will be high.

If input is between high and low limit then both high and low outputs will be zero. These outputs can be used as an alarm if input goes below low limit or above high limit.

Algorithm for High Low Compare is given as shown below:

```

If (Input >= HIGH_LIMIT) then
    High Output = 1;
    Low Output = 0;
else if (Input <= LOW_LIMIT) then
    High Output = 0;
    Low Output = 1;
else
    High Output = 0;
    Low Output = 0;
End if
    
```

G. HIGH LOW LIMITER

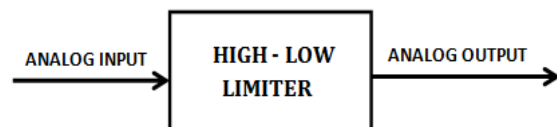


Figure 7: High Low Limiter Function Block

This block limits the input signal between two specified high limit and low limit. The analog output equals analog input

when the analog input is between the high limit and the low limit. The analog output equals the high limit when the analog input is higher than the high limit and the analog output equals low limit when the analog input is lower than the low limit.

Algorithm for this block can be given as:

```

if(Input >= HIGH_LIMIT) then
    Output = HIGH_LIMIT;
else if(Input <= LOW_LIMIT) then
    Output = LOW_LIMIT;
else
    Output = Input;
end if
    
```

H. ANALOG TRANSFER

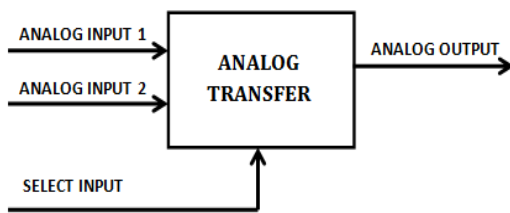


Figure 8: Analog Transfer Function Block

This block has two analog inputs (analog input1 and analog input2) one digital input (select input) and one analog output. If the select input is at logic 1 then analog output equals analog input1 and if the select pin is at logic 0 the analog output equals analog input2.

DIGITAL

Different digital logic blocks are used for manipulating the Boolean variables in the system. Most common among them are AND logic and OR logic. They are described below:

I. AND LOGIC

AND logic block gives high output only if all digital inputs to the block are high, else output is low in other case.

J. OR LOGIC

OR logic block gives high output if any of the digital inputs is high and low if all digital inputs are low.

DEVICE DRIVE LOGIC BLOCKS:

K. SOLENOID DRIVE

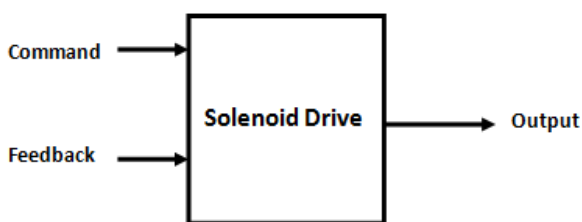


Figure 10: Solenoid Drive

In industries, solenoid valves are the most frequently used control elements in fluidics. Their tasks are to shut off, release, dose, distribute or mix fluids.so, to operate the solenoid valve, solenoid drive can be used. In which, command is given to the block and the output of the block causes relay pick up which in turn make valve open or close. Feedback is taken from the valve so as to confirm the status of the valve.

L. UNIDIRECTIONAL DRIVE

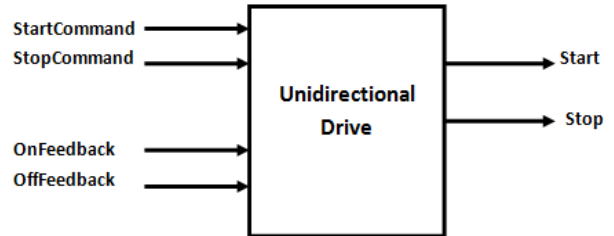


Figure 11: Unidirectional Drive

Motors are used for a wide range of industrial automation applications. Unidirectional drive is used to control the unidirectional motor (for example, fan or pump). Two commands for driving the motor - motor start command and motor stop command, because there are two separate contacts for on and off. So, when the command is given to the unidirectional drive, the relay contact is picked up and output device is operated. The feedback for the status of the output device is given back to the block and is monitored. If the feedback signal doesn't arrive within timeout value, the alarm is generated.

Algorithm for this drive can be derived as shown:

```

if(On Command == HIGH){
    On Output = HIGH;
    Off Output = LOW;
}else if(Off Command == HIGH){
    On Output = LOW;
    Off Output = HIGH;
}

Start Timer (Timeout Value);

While (Timer_is_ON){
    if((On Feedback == On Command) || (Off Feedback == Off Command)){
        Stop Timer ();
        Feedback Received = 1;
        Break;
    }
}

If(Feedback Received == 1){
    No alarm ();
}else{
    Generate alarm ();
}
    
```

M. BIDIRECTIONAL DRIVE

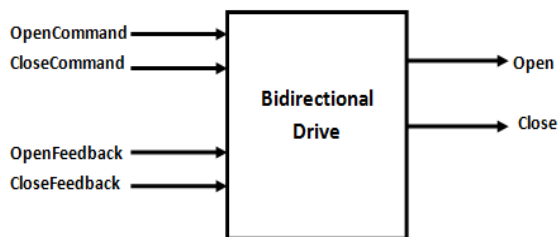


Figure 12: Bidirectional Drive

In bidirectional drive, there are two commands for driving the device in forward and reverse directions. When the open or close command is given to the bidirectional drive, the relay contact is picked up and output device is operated accordingly. The feedback for the status of the output device is given back to the block and the status of the feedback is monitored. If the feedback signal doesn't arrive within timeout value, the alarm is generated.

V. CONCLUSION

So, practical aspects of hardware and necessary interfacing circuits were studied and discussed in detail. Various functional blocks for control systems were studied in detail. Also, Algorithms of various functional blocks were prepared.

ACKNOWLEDGMENT

We are very much thankful to our Parents who continuously motivated and helped us in carrying out this new idea practically. We would like to thank our faculty for their support and guidance.

REFERENCES

- [1] Muhammad Ali Mazidi; *8051 microcontroller and embedded systems*; 2nd Edition, 2011.
- [2] Jacqueline Wilkie, Michael Johnson and Reza Katebi; *Control Engineering : An Introductory Course*; Palgrave Macmillan, 2001
- [3] Gayakwad. R.A, *Op-amps and linear integrated circuits*; 4th Edition, 2010
- [4] John.A.Shaw ; *PID Control Algorithm How it works, How to tune it and how to use it*; 2nd Edition, 2003
- [5] Richard Barnett, Larry O'Cull and Sarah Cox; *Embedded C Programming and the Atmel AVR*; Cengage Learning, 2006
- [6] Christian Diedrich, Francesco Russo, Ludwig Winkel ,Terry Blevins; *Function Block Application in Control System Based on IEC 61804*;
- [7] R.W. Lewis; *Programming industrial control systems using IEC 1131-3*; Revised Edition



Ashish. R. Patel is a student of Electronics Engineering in Maharaja Sayajirao University of Baroda. His areas of interests are VLSI, Software Programming and Embedded Systems



Preyas. T. Shah is a student of Electronics Engineering in Maharaja Sayajirao University of Baroda. His areas of Interests are mainly VLSI, Embedded Systems, Control Systems, Communication and robotics.