



REAL TIME LOGGING AND MONITORING OF ELECTROMECHANICAL MACHINES

Ashish V. Chandak

Shri Ramdeobaba College of Engineering and Management, Nagpur
Assistant Professor, Department of Information Technology
India

Purshottam J. Assudani

Shri Ramdeobaba College of Engineering and Management, Nagpur
Assistant Professor, Department of Information Technology
India

Naman Bajaj

Shri Ramdeobaba College of Engineering and Management, Nagpur
Graduate Student, Department of Information Technology
India

Abstract: Logging and monitoring the log generated, where logs generally tell a story of what has already happened, monitoring can tell a story about what is currently happening. There are many forms of monitoring. Some are more involved than others, but they all yield additional information about your environment. The industrial problem we are trying to solve is to ease the monitoring of electro-mechanical machines by logging the data generated by the machines and developing intelligent insights so as to manage the cost.

We have developed hardware and software applications which will help to log the data generated by the machines and monitor the output of the machines. We are using specific hardware to generate data from the machines. We are using the data generated by the machines to give detailed report to the owner of the industry. Report gives consolidated report of employee working on machine on daily basis. This will help to ensure that the given job is completed on time and the time requirement of the future job can be easily determined by visualizing the data generated previously onto the machines and thus productivity of resource person can be monitored.

Keywords: Arduino, Raspberry Pi, Logging.

I. INTRODUCTION

Traditional approach to save data in manufacturing industry on pen-paper does not guarantee the accuracy and consistency of the number of products generated by the particular worker on the machine. This leads to reporting of manipulated data to the owner of the industry. Thus leading to decrease in production and productivity for the industry. The following approach is currently being used by the industry for monitoring the productivity of resource person:

- Worker attendance is being taken manually on paper by the supervisor present on duty.
- The machine is allotted to the worker randomly without considering the proficiency of work done by the worker on the given machine.

This leads to decrease in production of the final product as compared to expected number of products. Hence this leads in the decrease in the profit share by the industry. There is no way to detect the number of product build on the particular machine by the worker. There is manipulation of the number of product build and attendance of the worker.

To overcome this problem, system has been designed to monitor the number of products produced by machine, and the time in which resource person is idle. We have designed an arduino based circuit that is mounted next to the PLC of machines and it will read the signals produced whenever the tool is put into the machine for processing. Data is fetched to computer based product raspberry PI stored and is stored data in database.

II. TOOLS AND TECHNOLOGY USED

A. Arduino

Arduino is an open-source computer hardware company that designs and manufactures microcontroller based kits for building digital devices and interactive objects that can sense and control the physical world [1]. Specification of Arduino Board is shown in Table 1.

Table 1. Specification of Arduino Board

Microcontroller	Atmega 328
Operating Voltage	5V
Digital I/O Pins	14
Analog Input	6
Flash Memory	32KB of which 0.5 is used by Boot loader
Clock Speed	16MHz
Width	53.4mm
Length	68.6mm

B. nRF24L01

The nRF24L01 is a highly integrated, ultra-low power (ULP) 2Mbps RF transceiver IC for the 2.4GHz ISM (Industrial, Scientific and Medical) band. The nRF24L01 integrates a complete 2.4GHz RF transceiver, RF synthesizer, and baseband logic including the Enhanced ShockBurst™ hardware protocol accelerator supporting a high-speed SPI interface for the application controller. No external loop filter, resonators, or VCO varactor diodes are required, only a low cost ±60ppm crystal, matching circuitry, and antenna. The nRF24L01 is available in a compact 20-pin 4 x 4mm QFN package [3].

Features:

- Low cost single-chip 2.4GHz GFSK RF transceiver IC
- Worldwide license-free 2.4GHz ISM band operation
- 1Mbps and 2Mbps on-air data-rate
- Enhanced ShockBurst™ hardware protocol accelerator
- Ultra low power consumption – months to years of battery lifetime
- On-air compatible with all Nordic nRF24L Series in 1 and 2Mbps mode
- On-air compatible with Nordic nRF24E and nRF240 Series in 1Mbps mode.

C. DS3231 RTC module

The DS3231 is a low-cost, extremely accurate I2C real-time clock (RTC) with an integrated temperature-compensated crystal oscillator (TCXO) and crystal. The device incorporates a battery input, and maintains accurate timekeeping when main power to the device is interrupted. The integration of the crystal resonator enhances the long-term accuracy of the device as well as reduces the piece-part count in a manufacturing line. The DS3231 is available in commercial and industrial temperature ranges, and is offered in a 16-pin, 300-mil SO package[2].

The RTC maintains seconds, minutes, hours, day, date, month, and year information. The date at the end of the month is automatically adjusted for months with fewer than 31 days, including corrections for leap year. The clock operates in either the 24-hour or 12-hour format with an AM/PM indicator. Two programmable time-of-day alarms and a programmable square-wave output are provided. Address and data are transferred serially through an I2C bidirectional bus.

A precision temperature-compensated voltage reference and comparator circuit monitors the status of VCC to detect power failures, to provide a reset output, and to automatically switch to the backup supply when necessary. Additionally, the RST pin is monitored as a pushbutton input for generating a µP reset.

III. SYSTEM DESCRIPTION

To monitor the CNC machine we decided to use cycle start button as our signal to check when the machine starts and when it is idle, to do so we required some mediator which can sense the button press signal, process it and send it to store in database for analysis purpose. For this we came with the architecture like creating a mediator circuit which can sense the signal and pass it to Raspberry PI which will act as Server for the mediator circuit.

We designed the mediator circuit based on commercial product of Arduino that can satisfy our requirements to sense and pass the signals to server. Our purpose was to just take

signals and send it to processor so we had the best option to use Microcontroller for mediator circuit, as the Microcontrollers are single task oriented. Microcontroller have some inbuilt memory EEPROM,ROM and flash memory hence these are good choice for real time monitoring as time is important constraint in Real time machines and Microcontroller’s features make them ideal for such system. In our project we used Atmel’s Atmega328 microcontroller which is an 8 bit microcontroller. For this mediator circuit we used following components [5].

Atmel Atmega328 (8 bit) Microcontroller

- LM7805 voltage regulator
- 22pf capacitor
- 22uf capacitor
- 10nf capacitor
- 1K ohm resistor
- 16MHZ crystal
- 28 pin dip socket
- Perfboard

On perfboard, we soldered the components based on the following circuit diagram

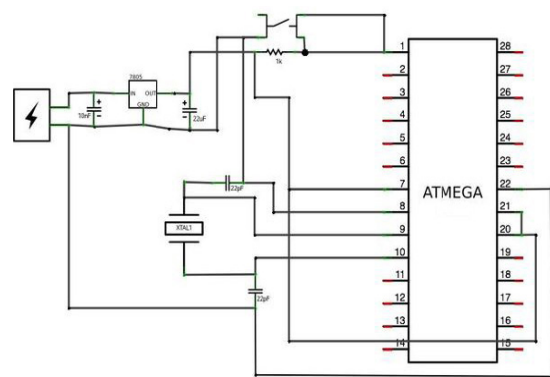
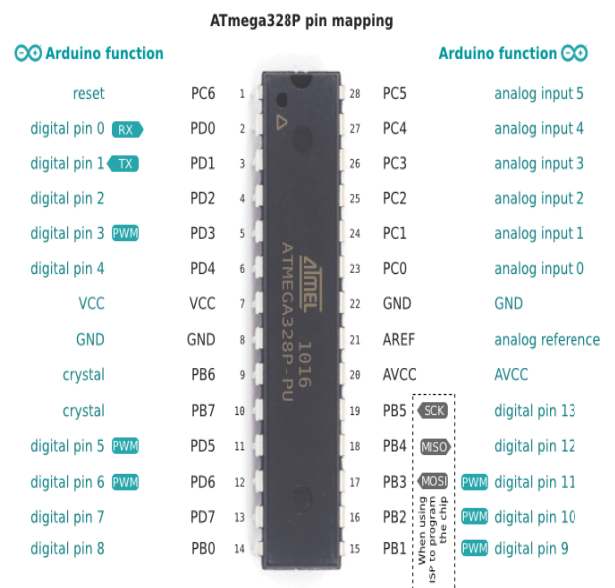


Fig. 1 Schematic of Arduino



A. Pin Description of Atmega328 Microcontroller

- Digital Pins: There are total 14 digital pins on atmega328 microcontroller, used for reading and writing digital signals from and to these pins.

- Serial: 0 (RX) and 1 (TX): Used to receive (RX) and transmit (TX) TTL serial data.
- External Interrupts: 2 and 3: These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value.
- PWM: 3, 5, 6, 9, 10, and 11: Provide 8-bit PWM output with the analogWrite() function.
- SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK): These pins support SPI communication.
- LED: 13: LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

B. Analog Pins:

There are 6 analog pins on atmega328 microcontroller which are used to connect to analog devices and to connect with other devices through wires using I2C protocol.

- I2C: 4 (SDA) and 5 (SCL): Support I2C communication using the Wire library

IV. SYSTEM ARCHITECTURE

We have created arduino based mediator and there is need of data transmission system between master and slave. So, we used serial transmission using USB ports but it has some drawbacks viz. expensive and tedious. So we used I2C protocol for communication.

Following architecture has been proposed for communication between raspberry and arduino based circuit:

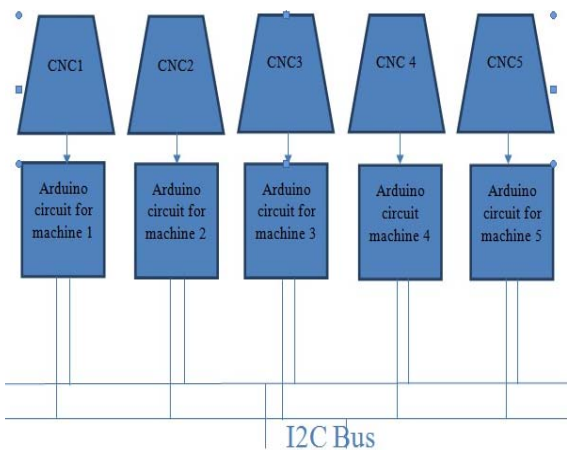


Fig 2 System Architecture

As shown in above figure 2, machines are connected to individual arduino circuits which are in turn connected to Raspberry pi via I2C Bus. With Raspberry Pi and I2C communication, we can connect the Pi with single or multiple Arduino boards. I2C can be used to connect up to 127 nodes which would be impossible using USB connections. It connects slaves via a bus that only requires two data wires, known as SDA and SCL.

Since the devices need a common voltage reference (Ground) and a power line the full bus has four wires, two for data and two for power. Sometimes additional lines are added, for example to support interrupts when the states of I2C devices change. The I2C bus can support multiple masters, but most micro-controllers can't. A master is usually a microcontroller, although it doesn't have to be. Slaves can be ICs or microcontrollers. When the master wishes to communicate with a slave it sends a series of pulses down the SDA and SCL lines. The data that is sent includes an address

that identifies the slave with which the master needs to interact. Addresses take 7 bits out of a data byte; the remaining bit specifies whether the master wishes to read (get data from a slave) or write (send data to a slave). Some devices have an address that is entirely fixed by the manufacturer; others can be configured to take one of a range of possible addresses. When a micro-controller is used as a slave it is normally possible to configure its address by software, and for that address to take on any of the 127 possible values. The address byte may be followed by one or more bytes of data, which may go from master to slave or from slave to master.

Each arduino will be assigned a unique address (slave address) like 0*00 the range of this addresses are from 00 to FF. This slave address assignment at arduino side makes it simple for raspberry to identify each arduino uniquely. The raspberry pi uses this slave address to send or receive data to and from the arduino.

A. Implementation of Cycle start counter:

As our prime goal was to sense, store and analyze cycle start signal first we focused on taking signal from machine so we decided to go with following approach as shown in figure 3.

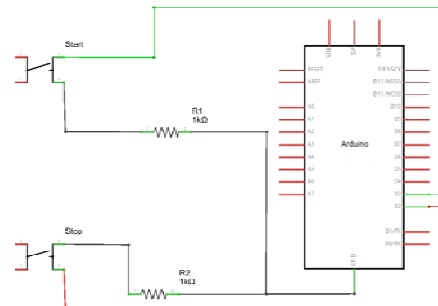


Fig 3 Schematic for sensing button press on Arduino

We studied the basic concept of button, the button always have at least two legs which are connected to power and ground respectively and the current only flows when these two legs are connected that is when the button is pressed then these two legs are connected and the circuit gets completed and the current passes through.

Similarly in the CNC machine there are two buttons cycle, start and cycle stop, with two legs as ground and two legs as power so we connected the arduino pin 2 with the ground pin to sense signal. so whenever the button is pressed the pin2 of arduino will have signal as high and when the button is released the pin will read low signal and by these whenever the pin2 reads the high signal we incremented the counter and passed it to raspberry [4].

As shown in fig. 3, we have connected the 1K ohm pull down resistor so in case when the button is not pressed the ground will always read low signal which implies the button is not pressed.

V. RESULTS

Logging table is created which stores data from Arduino. It contains following fields viz.

- Date: The date on which the data log was produced.
- Time: The time is taken in 24hrs format and the time is logged according to the products generated.
- Counter: The value of counter shows the number of products produced at any time

Table 2. Log Table

Date	Time	Counter
01-11-2017	7.00	1
01-11-2017	7.06	2
01-11-2017	7.13	3
01-11-2017	7.18	4
01-11-2017	7.26	5
01-11-2017	7.32	6
01-11-2017	7.36	7
01-11-2017	7.40	8
01-11-2017	7.44	9

Table 3. Log Table

Date	Machine Number	Counter
01-11-2017	1	143
02-11-2017	1	132
03-11-2017	1	120
04-11-2017	1	132
05-11-2017	1	115
06-11-2017	1	124
07-11-2017	1	110
08-11-2017	1	110

The above table shows the data logged on single machine. The data generated is as follows:

- Date: The date on which the log is generated.
- Machine Number: The machine number is the machine on which samples was generated.
- Counter: The counter is the value of the products generated at the end of the day on single machine.

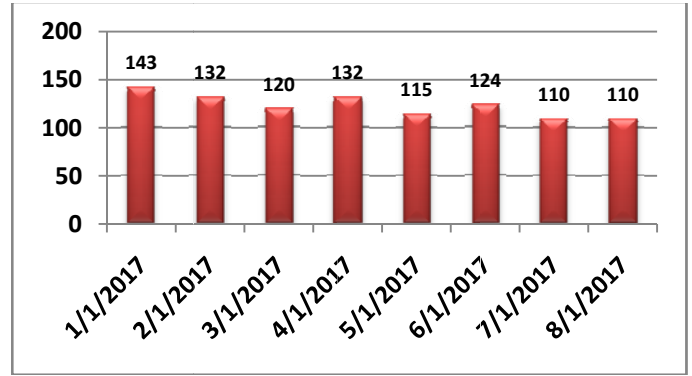


Fig 5 Visual Representation of Log

VI. CONCLUSION AND FUTURE WORK

We have developed the circuit for the single machine and in future we will design for 5 machines for logging and monitoring production. Wired network is not durable and cost effective and to overcome the problem of wired network we will developed wireless network for logging and monitoring production which uses NRF24101 wireless module to communicate with raspberry pi instead of using I2C protocol.

V. REFERENCES

- [1] <https://www.arduino.cc/en/Reference/HomePage>
- [2] https://en.wikipedia.org/wiki/Raspberry_Pi
- [3] <https://arduino-info.wikispaces.com/Nrf24L01-2.4GHz-HowTo>
- [4] <http://www.instructables.com/id/Perfboard-Hackduino-Arduino-compatible-circuit/>
- [5] <https://www.arduino.cc/en/Reference/HomePage>