Programmable Circuit Using PC's/Laptop's Motherboard

Professor S.R. Wattegaonkar [#1], Tejal Shinde [*2], Himanshu Singh [*3] , Hrithik Suri[*4]

[#]*Professor & Project Faculty member, Department Of Electronic & Telecommunication, Bharati Vidyapeeth College Of Engineering, Navi Mumbai, Maharashtra, India*

[*] *Student, Department Of Electronic & Telecommunication, Bharati Vidyapeeth College Of Engineering, Navi Mumbai, Maharashtra, India*

Abstract

With the advancement in automations it has somewhat become a norm nowadays for engineers/ technocrats of having a sound knowledge of robotics. It becomes very hard for students to implement their knowledge and create an interest until having a practical implementation. A practical implementation involves buying a basic board which will be programmed, it also requires the student to have a laptop for the same so it becomes tough to afford and at last many fail, so using the laptop/computer's motherboard if we are able to create programmable boards it would be very vital.

Keywords— *Robotics, IOT (Internet of Things), Motherboard*

I. INTRODUCTION

Today Raspberry PI plays a vital role in the field of robotics. The use of Raspberry PI is essential in the implementation of any idea related to robotics & IOT, but for testing purpose on a small scale everybody cannot afford Raspberry PI, since affording a Raspberry PI also involves requirements of some external devices such as keyboard, mouse, a display unit, etc. so this system can be used in replacement of a Raspberry PI in case of testing purpose on a small level.

The whole system-structure consists of a circuit designed on a PCB which involves the various PINS having their own function such as providing input to the main system or taking output from the system.

Mainly college going students aren't able to afford such a system which provides them the luxury of portability & at the same time lower the cost of buying one such board in which they can test their projects related to robotics & IOT, so such a system will solve both the issues. The system also provides the luxury to program in any language, which is somewhat limited in case of a Raspberry PI. One more advantage of such a system is that it will be more powerful at the same time provide system monitoring.

II. MOTIVATION

The motivation for building such a product came from self-experience & observing peers from our age group. Engineering students or those who are really interested in robotics really need such a system wherein they can test & implement their projects/ idea's practically, as this is such a field where practical implementation stands more essential than theoretical knowledge. All these happenings motivated us to come up with such a system which would be very useful in today's world.

III. METHODOLOGY

Here we look to send & receive signals via the USB port. USB is a technology where we can connect electronic devices to a computer. USB provides various advantages such as:-

- It is a plug & play system, i.e. does not require any restart
- Upto 127 devices can be connected
- Supports high speed transmission
- Length of the wire can be upto 5m

USB consists of four pins:-

- VDD
- DATA+
- DATA-

- GND

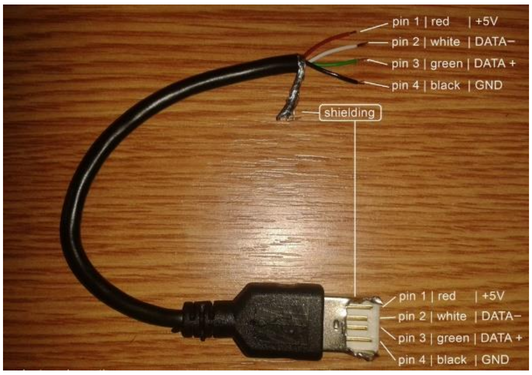

Fig. 1: Various PINS in USB (Image courtesy: https://www.electroschematics.com/)

The encoding scheme used in USB is NRZI (Non-return to zero interrupt). In NRZI one is represented by a no change in level & zero is represented by a change in level.

Bit Stiffing- When a series of zero's are transmitted in NRZI it causes transition of levels but when series of 1 is transmitted it causes no transition & this if carried for a long time confuses the receiver & makes it desynchronized. Bit stiffing is a process in which a zero is inserted after every six digits. In case if no transition takes place after six consecutive one's the receiver decides that no bit stiffing takes place and discards the data received.

SYNC field- Bit stiffing alone isn't enough to keep the host & receiver synchronized so SYNC field is also used. In this mechanism a SYNC field is used at start of a packet to enable the receiver to synchronize the clock.

The NRZI, Bit Stiffing & SYNC field are together used to synchronize the host & receiver.

Now after connecting the system to the computer we need to load the driver, for that the PID, i.e. Product ID & VID, i.e. Vendor ID is required we can obtain it by typing the 'lsusb' command.

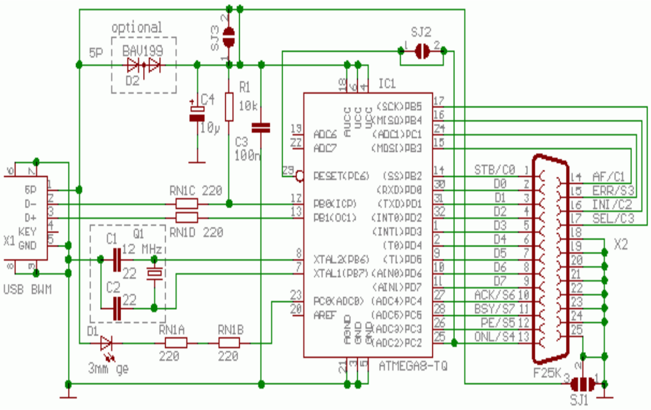

Fig. 2: Internal circuit diagram of parallel port (Image courtesy: www.user.tu-chemnitz.de)

The whole system-structure of the designed circuit involves a PCB with all the various pins mounted on it for various functions, viz. input & output.

Now this system is interfaced with the motherboard of the CPU which provides it memory & logic for any problem.

The system would be taking inputs from the connected devices such as sensors & convert them into suitable digital form & provide the input to the motherboard, now according to the logic the motherboard provides valid HIGH or LOW signals using its own internal registers. The Data, Control and status lines are connected to their corresponding registers inside the computer. So by manipulating these registers in program, one can easily read or write to parallel port using any programming language. Control register is connected to control lines and Status register is connected to Status lines. So whatever you write to these registers, will appear in corresponding lines as voltages, Of course, you can measure it with a multimeter & whatever you give to Parallel port as voltages can be read from these registers.

Sending DATA- The data can be sent to the USB port using the ‘echo command’ followed by the data to access the particular pin. In parallel port data is sent in HEXADECIMAL format. So, by using the particular value the various PINS can be accessed.

Receiving DATA- The signal given to the particular pin can be accessed by extracting the information in the file created by plugging in the particular USB device.

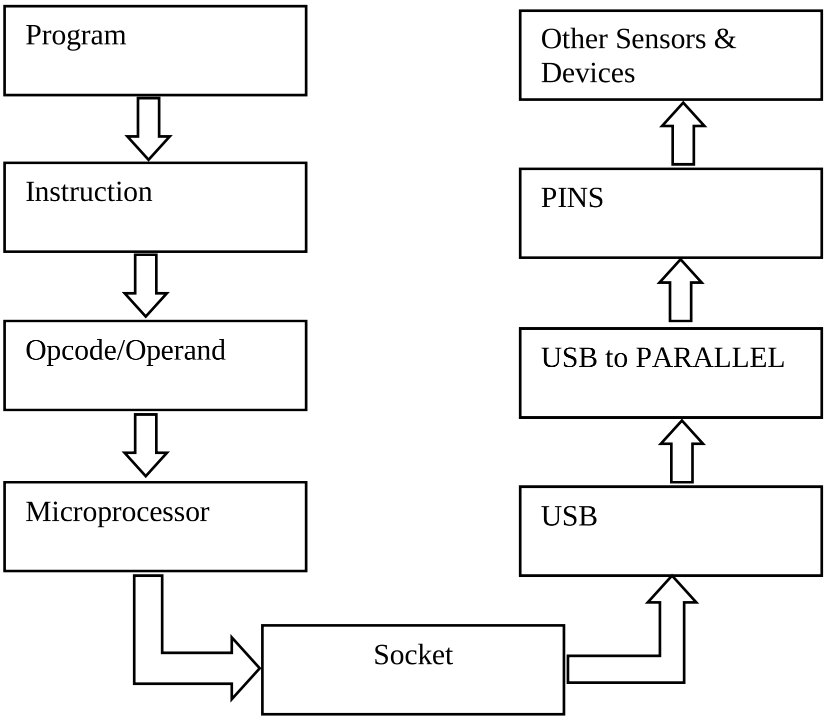

Fig. 3: Process Flow Schematic

IV. MAJOR COMPONENTS USED

The project contains a simple and small circuit consisting of the following components:

A. PCB

Printed Circuit Boards is also known as "printed wiring boards" or "printed wiring cards". It consists of 4 layers namely a base material or substrate which provides the durability next is a copper sheet/ above which is the soldermask layer, this gives the PCB its colour, the top layer is the silkscreen layer in which various indicators are used to indicate the function of each pin/ led.

B. PINS

The various pins that would be used for robotics purpose, i.e. taking input and giving output to the specified devices/sensors.

C. LPT CABLE DB36

This device contain parallel port type of interface for connecting different peripherals to the PC. It helps to send multiple bits of data at once. It provide multiple data lines and port connectors.

D.LAPTOP/DESKTOP

This would be used for providing the circuit memory & providing logic to solve any problem. It is responsible for the working of the whole system that we are trying to create.

V. CONCLUSION

In this paper, we make this model for young technocrats & engineering students who want to have a sound knowledge & understanding of robotics. For easy use our model would be connected to the Laptop/ Motherboard via USB cable which is available in every modern Laptop/ Desktop. For portability & convenience this system would be platform independent & we would be able to write the logic in any programming language.

REFERENCES

[1] Kansham Modar Maring, Kansham Angphun Maring, “Interfacing Parallel Port Using Turbo C Programming”,(July 2015),International Journal of Scietific & Research Publications(IJSRP)

[2] A. Sandanasamy, “Exploring DB25 as an Interface to Electronic Appliances & Automating Remotely with SMS”,(Jan 2015),International Journal of Science, Engineering & Technology Research(IJSETR)

[3] Kim I. Mallalieu, Ronald Arieatas, “Internet-Enabled Parallel Port Driver”,(1999),(TEMPUS)

[4] IEEE Standard Signaling Method for a Bidirectional Parallel Peripheral Interface for Personal Computers, IEEE Std 1284 2000(R2011), (Revision of IEEE Std 1284-1994)