International Journal of Engineering Research in Electronic and Communication Engineering (IJERECE) Vol 3, Issue 5, May 2016

Industrial Iot Implementation Using Scanner Based Monitoring

[1] Ashwin Jose [2] Karthik Bekal [3] Kavita C [4]Khomal Kalyan Kumar[5]Jayaraj N
[1][2][3][4] Scholar, [5] Professor,
[1][2][3][4][5] Department of ECE,
The Oxford College of Engineering Bangalore, Karnataka, India

Abstract— **The Raspberry Pi is a miniature computer that can be used in electronic projects. The Pi's functionality ranges over quite a lot of domains; this can be exploited to entirely replace different devices to interface Human Interfaces or Programmable Logic Controllers (PLC). In our project we explore the idea of a raspberry Pi functioning as a standalone client computer, which interfaces with a ttl interfaced Barcode scanner while communicating with a server. The Pi also interfaces with different interfaces to help the operator interact with the client side computer. The Pi has sufficient computing power to behave as a PLC to control the various testing equipment and control machinery used in the production line, where this project was initially conceived to be implemented.**

I. INTRODUCTION

The project was initially conceived to optimize space in a production line. The raspberry pi proved quite useful as it is a computer the size of an average credit card. The objective was to have the Pi replace a fully functional Industrial Computer. This would mean to incorporate functions to have the Pi function as an interface to the many devices used in a production line for example a Barcode Scanner. The Pi also has to communicate with the main server which tracks and stores all information about the products running through all the lines in the Production Floor. We can also implement a database to track the products that passed through the line, hence a better tracking process can be implemented with the main server having data of every product that was processed in the production line, and the pi having information about each and every product that was processed in that particular line. These being the basic implementation, we can further expand the space conservation policy and provide dedicated buttons for inputs and LED's for responses.

II. UNDERSTADING THE REQUIREMNTS

a. *Problem Statement*

- ❖ The current controllers used in the production line lack TCP/IP protocols and hence can't be used to connect to the main server.
- ❖ Using a Raspberry Pi we can build a standalone client computer that covers minimal space but it can also act as a Programmable Logic Controller
- ❖ The use of space must be optimized as testing lines in the production line might not have space to accommodate a full-fledged Industrial Computer.
- ❖ The raspberry pi has pins that have a direct connection with the CPU of the PI, this helps in interfacing legacy devices and also new Human Interfaces which can help the operators function optimally.

The raspberry pi 2 is a credit card sized computer which is a capable little computer which can be used in electronic projects and for many of the things that your desktop PC does. Hence, use of a Pi to conserve space while maintaining full functionality is preferred.

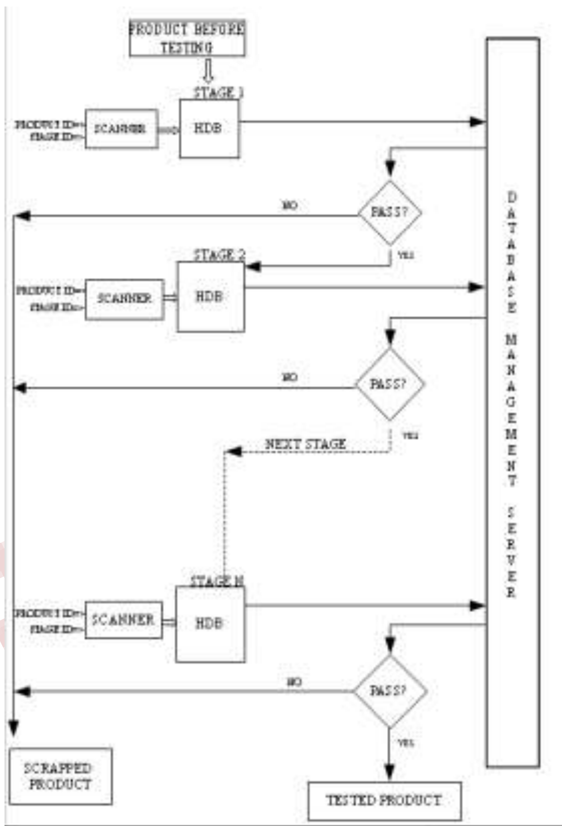
b. Objectives

The project consists of three sections; the first one is interfacing the barcode scanner with a Raspberry Pi to scan the product barcode. The barcode of the product when scanned provides essential data such as the product id, product status and stage id. This allows continuous monitoring of the product in production line.

In the second section, the essential data obtained from the line specific barcode is sent to the main server over a LAN interface. The main server provides sufficient data management with security via cloud storage.

In the third section, the positive or negative test result of the product is sent from the main server to the respective stage with minimum time consumption. After which the product is either passed over to the next stage or attempts are made to recover the product, if neither then the product is scrapped. The above three sections are repeated for each stage in the production line that involves testing.

Then the pi can be programmed to behave as a PLC while still handling all the client side activities. The dedicated pins on the Pi can be used to interface any type of legacy device or even modern devices like Touch screen interfaces. The dedicated pins can be used to provide a minimalistic interface by having buttons behave as inputs to the client side and LED's to show responses from the main server and the client side program.



Block Diagram

Fig.1. Block diagram of production line.

This diagram hopes to explain the implementation in a production line. Consider the product entering the stage 1 of the production line. To identify this product we use barcodes, which are produced based on a specified template. It is essential to track the product for error correcting purposes in manufacturing and also to provide detailed information about the tests carried out each stage.
This entire process is already implemented using an industrial computer which occupies tremendous amount of space. Here we make use of a mini computer called the Raspberry Pi. The Pi occupies minimal space compared to the industrial computer, as it is only the size of an average credit card.
The process starts with the product entering stage 1; here the product ID is retrieved from the barcode already placed on the product.

Depending on the type of hardware used for barcode scanning, different encoding methods are used to retrieve data by the Pi. The Pi also contains stage related information which is essential for tracking. Once the data has been received and decoded it is packaged or formatted to be sent to a server. This server contains information regarding each product at every stage. The Pi also passes along information regarding the status of the product such as 'Pass' or 'Fail' in a certain test cycle.

Once the product passes on to the next stage, the Pi then checks for 'Pass' or 'Fail" status from the previous stage. If the product failed the previous stage, rectifications are made or else it is scrapped.

III. FLOW DESCRIPTION

The language used here for programming is Python. Python is a general-purpose interpreted, interactive, object-oriented, and high-level programming language. It was created by Guido van Rossum during 1985- 1990. Like Perl, Python source code is also available under the GNU General Public License (GPL). Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

The first step of the process is to initialize the serial port. In this flow chart the serial port represents the software extension to a device. A serial port has parameters such as baud rate, parity, stop bits and data bits. By specifying the parameters, we can establish a serial line connection from the Raspberry Pi to the barcode scanner. After initializing this serial port, the user has the provision to change the parameters of the serial port. Once the parameters have been finalized the serial port is opened and data can now be received serially. After the product barcode is scanned, the barcode is converted to a user specified data format. This data is available in the serial port buffer. After the Pi successfully receives the data it is sent to the server along with the stage ID that the Pi is in.

Fig. 2 .Flow chart for client side program

The Pi communicates with the server over a wired LAN interface while using TCP\IP protocol. Before sending the data, the Pi has to format the barcode data and stage data into server specific strings. The Pi also performs basic connection checks to let the user know that there is already an active connection between the Pi and the server. It is ideal to send a start bit at the start of the data frame and a stop bit at the end of the data frame to ensure that the server receives the data with no error.

The server maintains a data base containing information related to each product at every stage. The received data is decoded and compared with the previous status of the product in the last stage. The server then informs the Raspberry Pi if the product has passed the last stage. Based on this information the Pi sends the product to the next stage or for it to be scrapped.

A GUI is developed for easy interaction between the operator and the Pi. We can also implement alerts from the Pi as visual indications using LEDs. The Pi's general purpose input/output pins can be used access the other hardware devices like scanners, PLCs and sensors. The Pi can also function as a stand-alone Programmable Logic Controller to control test equipment used in the production line.

IV. RETURN ON INVESTMENT

From a Managements perspective any new implementation must have a highly positive ROI (Return on Investment).

So let us consider the ROI for a raspberry pi implementation.

In terms of "Man Power", this implementation doesn't affect much of the process. The same amount of physical labor is required before and after the implementation. There is a need for an operator to overlook each station and since this implementation isn't completely autonomous, the labor requirement remains the same.

The "Machine" here has a radical shift from an Industrial PC to a Mini Computer, i.e. the Raspberry Pi. This shift costs the management much lesser, the drawback here is the Mini PC is limited in terms of expandability, since the raspberry pi has limited functionality.

The "Method" of the process for tracking remains the same, hence the time required to implement this is same and or lesser than the time to implement the process on an industrial PC.

"Measurements" performed by the Raspberry Pi implementation is the same as industrial PC hence, the error is almost minimal or even acceptable.

Finally, to calculate the cost saved let's look at a few areas of concern:

- Time: Implementation on the Pi is lesser since its Operating System can be cloned in a few minutes and hence multiple Pi's can be put into use in a few minutes.
- Space: The space a Raspberry Pi can fit inside a person's pocket and hence a lot of space can be saved using the Pi compared to an Industrial PC.
- Man Power: There is no improvement or regression in terms of Man Power.
- Rejection: The rejection ratios are still controlled in the production process and not the monitoring process, Hence, neither the Pi nor the Industrial Computer influence the Rejections in the production line.

Overall, we can estimate the amount of implementation for a Raspberry Pi to be around Rs.3500/-, whereas upgrading an Industrial Pc or a PLC to have the same functionality can be estimated to Rs.1, 00,000/-, which is a substantial amount of saving.

REFERENCES

[1] The Glasgow Raspberry Pi Cloud: A Scale Model for Cloud Computing Infrastructures - Fung Po Tso. In Distributed Computing Systems Workshop (ICDCSW), 2013 IEEE 33rd International Conference on July 2013.

[2] Design and implementation of enterprise asset management system based on IOT technology- Minli Wang. In Communication Software and Networks (ICCSN), 2015 IEEE International Conference on 6-7 June 2015.

[3] Research and application of TCP/IP protocol in embedded system. In Communication Software and Networks (ICCSN), 2011 IEEE 3rd International Conference on 27-28 May 2011

[4] Python Programming http://www.tutorialspoint.com/python/python_classes_objects.htm