International Journal For Advanced Research In Science & Technology
A peer reviewed international journal
www.ijarst.in
ISSN: 2457-0362

Water level meter for alerting population about floods

B. Sai Prasanna[1], C Ramakrishna Reddy[2]
Assistant Professor[1,2]
Department OF ECE
Malla Reddy Engineering College

Abstract- The maximum crucial component without delay before, at some point of and after a catastrophe takes place is the dissemination of records, a deployment of gadgets enabled through IoT (Internet of Things) may want to convey blessings in phrases of giving to human beings records opportunely for making choices in face of this catastrophe. In this paper, we gift a sensor to degree water degree in rivers, lakes, lagoons and streams. For such motive and to show our concept, we designed a pilot challenge thru a micro-version this is built with a water degree size sensor primarily based totally on a easy open circuit that closes while in touch with water and experimentally examined right into a water field beneathneath a managed surroundings. This micro-version is executed on the idea of a programmable digital board (Netduino Plus 2), an digital circuit related to electric resistances which are positioned at a particular height, inside a water field, while the water degree rises and reaches the resistors, varies the impedance, this indicates the real water degree and so forth for distinctive heights. The records from water degree sensor is transmitted through WiFi to a laptop, then this records is likewise visible in smartphones, wherein customers can see the water degree in rivers. Finally, the micro-version is examined through experimental exams beneathneath a managed surroundings and great consequences are obtained.

Index Terms- IOT, Water Level Meters, Floods

I. INTRODUCTION

Automated sensors and remote communication aims at building a highly automated dam gates controlling system by making use of sensors like water level indicators, gate status monitoring on LCD, gate control using DC motor and alerting system. The objective of this project is fulfilled by employing a water level detection sensor for water level indication, gate position monitoring and control system, and remote communicating using Wi-Fi module to the user Android phone. The sensors altogether form the input module of the project. The project makes use of microcontroller, which acts as a central controlling unit. This module is capable of communicating with the input and the output modules. The output module is formed by the motors used for closing and opening of gates and also alerts to the mobile phones with the Wi-Fi module. The micro controller reads the sensor inputs continuously.

The major aim of this system is to provide remotely a caution alarming message using Wi-Fi module to the user Android phone. Water level sensor keeps track of the water level in the dam. The system operates the dam gates using DC motors automatically when water level in the dam crosses the threshold value. The system consists of a Wi-Fi module, Water level sensors, LCD, buzzer. Water level and status of the dam gates are displayed on the LCD. When the dam gates get opened, alert is given in the form of buzzer and message goes to the user Android phone through Wi-Fi module.

The main aim of this project is to provide an efficient solution for monitoring and control of dam water gates with automatic gate control using DC motors. This system also enables the alerting system using Wi-Fi module to the user Android phone.

An embedded system is a combination of software and hardware to perform a dedicated task. Some of the main devices used in embedded products are Microprocessors and Microcontrollers.

Microprocessors are commonly referred to as general purpose processors as they simply accept the inputs, process it and give the output. In contrast, a microcontroller not only accepts the data as inputs but also manipulates it, interfaces the data with various devices, controls the data and thus finally gives the result.

The project"Water Level Meter for Alerting Population about Floods" using PIC16F72 Microcontroller is an exclusive project that can control the dam gates based on the water levels in the dam.

II. PROPOSED WORK

PIC stands for Peripheral Interface Controller given by Microchip Technology to identify its single-chip microcontrollers. These devices have been very successful in 8-bit microcontrollers. The main reason is that Microchip Technology has continuously upgraded the device architecture

International Journal For Advanced Research In Science & Technology
A peer reviewed international journal www.ijarst.in
ISSN: 2457-0362

and added needed peripherals to the microcontroller to suit customers' requirements. The development tools such as assembler and simulator are freely available on the internet at www.microchip.com .

Low - end PIC Architectures:

Microchip PIC microcontrollers are available in various types. When PIC microcontroller MCU was first available from General Instruments in early 1980's, the microcontroller consisted of a simple processor executing 12-bit wide instructions with basic I/O functions. These devices are known as low-end architectures. They have limited program memory and are meant for applications requiring simple interface functions and small program & data memories. Some of the low-end device numbers are

12C5XX
16C5X
16C505

Mid range PIC architectures are built by upgrading low-end architectures with more number of peripherals, more number of registers and more data/program memory.

Popularity of the PIC microcontrollers is due to the following factors.

1. Speed: Harvard Architecture, RISC architecture, 1 instruction cycle = 4 clock cycles.
2. Instruction set simplicity: The instruction set consists of just 35 instructions (as opposed to 111 instructions for 8051).
3. Power-on-reset and brown-out reset. Brown-out-reset means when the power supply goes below a specified voltage (say 4V), it causes PIC to reset; hence malfunction is avoided.

A watch dog timer (user programmable) resets the processor if the software/program ever malfunctions and deviates from its normal operation.

4. PIC microcontroller has four optional clock sources.
   - Low power crystal
   - Mid range crystal
   - High range crystal
   - RC oscillator (low cost).
5. Programmable timers and on-chip ADC.
6. Up to 12 independent interrupt sources.
7. Powerful output pin control (25 mA (max.) current sourcing capability per pin.)
8. EPROM/OTP/ROM/Flash memory option.
9. I/O port expansion capability

SUBROUTINES

A subroutine is a section of code, or program, than can be called as and when you need it. Subroutines are used if you are performing the same function more than once, for example creating a delay. The advantages of using a subroutine are that it will be easier to alter the value once inside a subroutine rather than, say, ten times throughout your program, and also it helps to reduce the amount of memory your program occupies inside the PIC.

First, we have to give our subroutine a name, and in this case we have chosen ROUTINE. We then type the code that we want to perform as normal. In this case, Wehave chosen the delay in our flashing led program. Finally, we end the subroutine by typing the RETURN instruction.

To start the subroutine from anywhere in our program, we simply type the instruction CALL followed by the subroutine name.

Let us look at this in slightly more detail. When we reach the part of our program that says CALL xxx, where xxx is the name of our subroutine, the program jumps to wherever the subroutine xxx resides. The instructions inside the subroutine are carried out. When the instruction RETURN is reached, the program jumps back to our main program to the instruction immediately following our CALL xxx instruction.

You can call the same subroutine as many times as you want, which is why using subroutines reduces the overall length of our program. However, there are two things you should be aware of. First, as in our main program, any constants must be declared before they are used. These can be either declared within the subroutine itself, or right at the start of the main program. We would recommend that you declare everything at the start of your main program, as then you know that everything is in the same place. Secondly, you must ensure that the main program skips over the subroutine. What We mean by this is if you put the subroutine right at the end of your main program, unless you use a ‘Goto’ statement to jump away from where the subroutine is, the program will carry on and execute the subroutine whether you want it to or not.

3.7 Serial communication

Beside stated above we've added to the already existing unit the possibility of communication with an outside world. However, this way of communicating has its drawbacks. One of the basic drawbacks is the number of lines which need to be used in order to transfer data. What if it is being transferred to a distance of several kilometers? The number of lines times’ number of kilometers doesn't promise the economy of the project. It leaves

International Journal For Advanced Research In Science & Technology
A peer reviewed international journal
www.ijarst.in
ISSN: 2457-0362

us having to reduce the number of lines in such a way that we don't lessen its functionality. Suppose we are working with three lines only, and that one line is used for sending data, other for receiving, and the third one is used as a reference line for both the input and the output side. In order for this to work, we need to set the rules of exchange of data.

These rules are called protocol. Protocol is therefore defined in advance so there wouldn't be any misunderstanding between the sides that are communicating with each other. For example, if one man is speaking in French, and the other in English, it is highly unlikelythat they will quickly and effectively understand each other. Let's suppose we have the following protocol. The logical unit "1" is set up on the transmitting line until transfer begins. Once the transfer starts, we lower the transmission line to logical "0" for a period of time (which we will designate as T), so the receiving side will know that it is receiving data, and so it will activate its mechanism for reception. Let's go back now to the transmission side and start putting logic zeros and ones onto the transmitter line in the order from a bit of the lowest value to a bit of the highest value. Let each bit stay on line for a time period which is equal to T, and in the end, or after the 8th bit, let us bring the logical unit "1" back on the line which will mark the end of the transmission of one data. The protocol we've just described is called in professional literature NRZ (Non-Return to Zero).

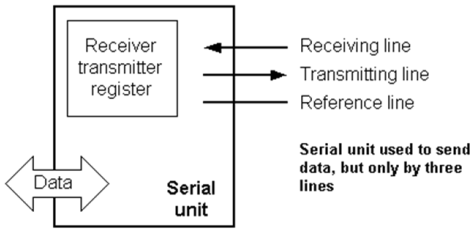


Fig 3.4: serial unit

As we have separate lines for receiving and sending, it is possible to receive and send data (info.) at the same time. So called full-duplex mode block which enables this way of communication is called a serial communication block. Unlike the parallel transmission, data moves here bit by bit, or in a series of bits what defines the term serial communication comes from.

After the reception of data we need to read it from the receiving location and store it in memory as opposed to sending where the process is reversed. Data goes from memory through the bus to the sending location, and then to the receiving unit according to the protocol.

WATCHDOG TIMER

Suppose you have written a program that is continuously running on a PIC. Now, you want to make sure that this program is always running, and that no matter what happens it will never stop. The first thing you would have, of course, is a loop back at the end of the program that brings us back to the start of the program. But consider this case. Let us say that the PIC is monitoring an input. When this input goes high, it jumps to another part of the program and waits for another pin to go high. If the second pin doesn't go high, the PIC will just sit there and wait. It will only exit if the second pin goes high. Let us consider another example. Suppose you have written a program. You have compiled it successfully, and you have even simulated it over and over again using a simulator such as MPLAB. Everything seems to work fine. You program the PIC and place it into a circuit. However after a long period of time, the program gets stuck somewhere and the PIC gets caught in a loop. What's needed in both cases is some kind of reset if the program gets stuck. This is the purpose of a watchdog circuit.

A watchdog circuit is nothing new. Many microprocessors and microcontrollers have them. But how does it work? Well, inside the PIC there is a resistor/capacitor network. This provides a unique clock, which is independent of any external clock that you provide in your circuit. Now, when the Watchdog Timer (abbreviated to WDT) is enabled, a counter starts at 00 and increments by 1 until it reaches FF. When it goes from FF to 00 (which is FF + 1) then the PIC will be reset, irrespective of what it is doing. The only way we can stop the WDT from resetting the PIC is to periodically reset the WDT back to 00 throughout our program. Now you can see that if our program does get stuck for some reason, then the WDT will not be set. The WDT will then reset the PIC, causing our program to restart from the beginning.

In order to use the WDT, we need to know three things. First, how long have we got before we need to reset the WDT, secondly how do we clear it. Finally, we have to tell the PIC programming software to enable the WDT inside the PIC. Let's look at these separately.

WDT Times

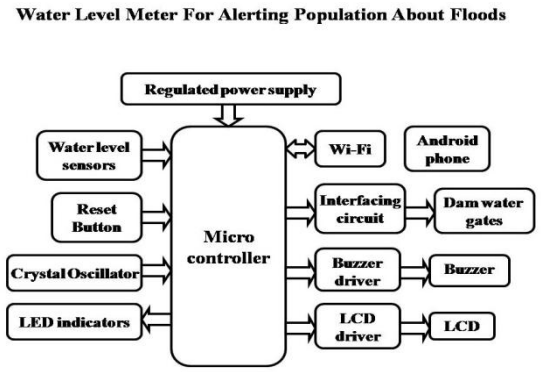
The PIC data sheet specifies that the WDT has a period from start to finish of 18mS. This is dependant several factors, such as the supply voltage, temperature of the PIC etc. The reason for the approximation is because the WDT clock is supplied by an internal RC network. The time for an RC network to charge depends on the supply voltage. It also depends on the component values, which will change slightly depending on their temperature. So, for the sake of simplicity, just take it that the WDT will reset every 18mS. We can, however, make this longer. Inside the PIC is a thing called a Prescaler. We can program this

International Journal For Advanced Research In Science & Technology
A peer reviewed international journal
www.ijarst.in
ISSN: 2457-0362

prescaler to divide the RC clock. The more we divide the RC clock by, the longer it takes for the WDT to reset.

The prescaler is located in the OPTION register at address 81h, bits 0 to 2 inclusive. Below is a table showing the bit assignments with the division rates and the time for the WDT to time out:

Bit 2,1,0	Rate	WDT Time
0,0,0	1:1	18mS
0,0,1	1:2	36mS
0,1,0	1:4	72mS
0,1,1	1:8	144mS
1,0,0	1:16	288mS
1,0,1	1:32	576mS
1,1,0	1:64	1.1Seconds
1,1,1	1:128	2.3Seconds

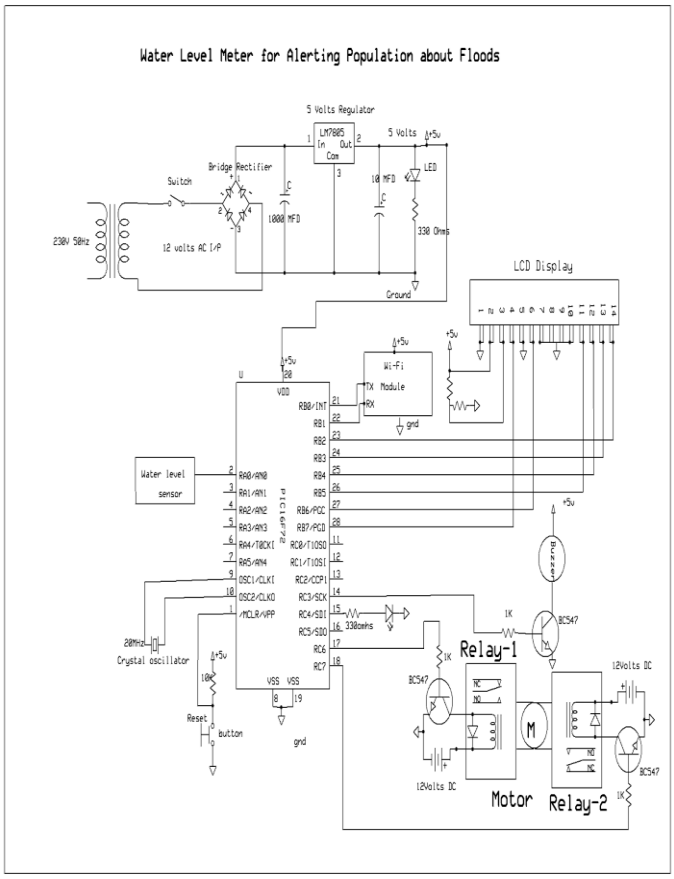


Block diagram of the project

The schematic diagram of Water Level Meter for Alerting Population about Floods explains the interfacing section of each component with microcontroller. The crystal oscillator connected to 9th and 10th pins of micro controller and regulated power supply is also connected to micro controller and LED's also connected to micro controller through resistors and motor driver connected to micro controller. Water level sensor keeps track of the water level in the dam. The system operates the dam gates

The project "Water Level Meter for Alerting Population about Floods" is to provide an efficient solution for monitoring and control of dam water gates with automatic gate control using DC motors. This system also enables the alerting system using Wi-Fi module to the user Android phone.

using DC motors automatically when water level in the dam crosses the threshold value. The system consist of a Wi-Fi module, Water level sensors, LCD, buzzer. Water level and status of the dam gates are displayed on the LCD. When the dam gates get opened, alert is given in the form of buzzer and message goes to the user Android phone through Wi-Fi module.



Schematic diagram of water level meter for alerting population about floods

it is the time to articulate the research work with ideas gathered in above steps by adopting any of below suitable approaches:

III. RESULTS AND DISCUSSION

IV. CONCLUSION

Integrating features of all the hardware components used have been developed in it. Presence of every module has been reasoned out and placed carefully, thus contributing to the best working of the unit. Secondly, using highly advanced IC's with the help of growing technology, the project has been successfully implemented. Thus the project has been successfully designed and tested

International Journal For Advanced Research In Science & Technology
A peer reviewed international journal
www.ijarst.in
ISSN: 2457-0362

REFERENCES

D. Giusto, A. Iera, G. Morabito and L. Atzori, The Internet of Things, Springer-Verlag, 2010

L. Atzori, A. Iera and G. Morabito, "The internet of things: A survey", Computer Networks, vol. 54, no. 15, pp. 2787-2805, 2010.

D. Miorandi, S. Sicari, F. D. Pellegrini and I. Chlamtac, "Internet of things: Vision applications and research challenges", Ad Hoc Networks, vol. 10, no. 7, pp. 1497-1516, 2012.

J. Gubbi, R. Buyya, S. Marusic and M. Palaniswami, "Internet of things (iot): A vision architectural elements and future directions", Future Generation Computer Systems, vol. 29, no. 7, pp. 1645-1660, 2013.

W. Kang and Y. Shibata, "Performance Evaluation of Disaster Information System Based on P2P network", Advanced Information Networking and Applications Workshops (WAINA) 2010 IEEE 24th International Conference on, pp. 710-715, 20–23 2010.

J. Kim, D. Kim, S. Jung, M. Lee, K. Kim, C. Lee, et al., "Implementation and Performance Evaluation of Mobile Ad Hoc network for Emergency Telemedicine System in Disaster Areas", Engineering in Medicine and Biology Society 2009. EMBC 2009. Annual International Conference of the IEEE, pp. 1663-1666, 3–6 2009.