Instinctive Gas Booking System by Using Intel Galileo Gen 2

T.Narendra kumar[1], M.Sravanthi[2], SK.Dilshad[3]
Assistant Professors, Department of ECE
CMR Technical Campus, Hyderabad, India
tanneru.narendra903@gmail.com[1],matams62@gmail.com[2],skdilshad.ece@gmail.com[3]

Abstract: Residual fuel detection in LPG cylinder is a major problem in domestic applications and gas driven vehicles. The best way to know about it is by using a continuous weight monitoring system of the cylinder .we here present a system which monitors the completion of the fuel in the cylinder and also booking the refill automatically with the LPG provider. This design has a load cell, which measures the weight of the gas cylinder and produces a current signal, proportional to the weight. This current signal is converted to an amplified form of voltage signal by the signal conditioning block, which is then compared with a threshold value. A sensor detects the error signal and sends the information to control unit, which triggers the control unit to send appropriate messages using GSM module. The design also alerts the consumer by sending a message through a GSM Module whenever the fuel level in the gas cylinder is lower than a threshold level.

Index Terms: Intel Galileo Gen 2, load cell, INA125P Amplifier, GSM module

I.INTRODUCTION

Many a times it is difficult to monitor the fuel level in gas cylinder and estimate the remaining fuel to make an advance booking with LPG provider. In today’s scenario of busy lives and preoccupied schedules people even forget to make the booking. Our system provides the solution in such cases. This is a prototyping example of a real time working module. The key objective is to develop a low cost system which can monitor the level (Weight) of the cylinder in the particular household and provide an alert in case of cylinder being empty. Being safe and secure is the need of the hour, so our effort behind this system is to design and fabricate a gadget which is so compact in itself that provides advantage of personal monitoring system. This device will be very useful for the household. This will prove as a multi-pronged strategy with the participation of multi stake holders of the society. The device is highly sensitive and easy to handle. Its response will provide information and monitoring of cylinder to individual user.

An Intel Galileo is the updated version of an Arduino Uno board, little bit larger compared to Arduino Uno, has a powerful processor, increased memory capacity for storage of code and data, has a facility to get connected to an Ethernet network and a USB port to connect computer accessories or the Mini PCI Express connector at the bottom of the board. Intel’s Quark SoC X1000 is the processor at the heart of the Intel Galileo. The firmware that runs on the Galileo is much more advanced than what’s on an Arduino Uno. On a Uno and most other Arduino devices, there is firmware on the board called the boot loader which is meant to help you upload and run your code on the board’s processor. It only does that and not much else. The firmware on the Galileo, on the other hand, is much more advanced. Not only does it help you upload and run your code on the board, but it also keeps track of files, the date and time of day, and helps share the board’s various resources between multiple programs running at the same time. In that way, it’s more like a typical computer. In fact, the firmware on the Galileo is a version of Linux, the free operating system that powers many desktop computers and servers these days. Galileo may not have a screen or desktop environment, but it still has much of the functionality that an operating system affords. And through your Arduino code, you’ll be able to access this functionality, giving you much more facilities than you’d have with a typical Arduino. For instance, if you want your project to take a picture from a web cam and email it, it’s something that would be difficult to do with only Arduino code. But with the power of Linux, this could be done more easily.

II.PROPOSED WORK

A pressure sensor/weight scale sensor is placed under the gas cylinder. It measures the weight of the cylinder and pressure exerted by the gas cylinder on to the ground. Experimental values are recorded from pressure sensor/weight scale sensor at proper time intervals. These values are sent to the Intel Galileo board. By setting up a weight limit/pressure limit from the recorded experimental values, when the gas in the cylinder reduces than the limit value, intimation is given to the consumer mobile number.

These processes are done by interfacing Intel Galileo with GSM module used for sending messages. The GSM module sends a message to the Gas provider. The message consists of the data required for Gas booking. The refill is booked and intimation is send to the customer about details of booking.

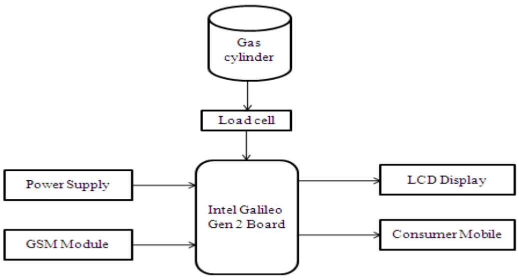


Fig. 1 System Overall Block Diagram

A. INTEL GALILEO GEN 2

Galileo is a an electronic circuit board, that helps you develop interactive objects by reading information from the world outside, process it, and then take action in the real time. If it's connected to a network, it has the ability to communicate with other devices like web servers. Galileo is an Arduino-compatible development board. The purpose of the hardware and software in the Arduino platform is to reduce complexity while designing an electronic system. It's meant to let you experiment, invent, and explore creative uses of technology with an ease of doing it and avoiding the need of technical mastery. By offering compatibility with Arduino hardware and software, Intel Galileo delivers an easy-to use platform but has more power and features than typical Arduino boards. Similar to the available development boards, Galileo has the capability to control and modify the outputs based on the inputs given. An input brings information from the physical world into the board's processor. Sensors can also be used as inputs to read information from the board.There are plenty of sensors to choose from including velocity, acceleration, temperature, pressure, sound level, brightness, and much more. The output shows how the Galileo development board can affect the physical world. It can be a simple LED, indicating the status (on/off) of the device, or the occurrence of an error in the code (a blinking red LED would be perfect for that). Outputs could be a speaker that plays musical tones, a text display for the pressure on an LCD screen, or servo motors that drive a robot etc.,

The Galileo can also communicate with other devices in a few different ways. You can connect it to your computer through an USB for transmission of data. Galileo sends information about the code being executed to a console window running on your computer so that you get a clear view of why something isn't working right and debug it. You can even have it send information about sensors to the computer and display a live graph. Galileo can also connect to other devices over the Internet using its built-in Ethernet or an optional Wi-Fi module. It can receive information about the weather or your email. It can search Twitter and much more. It can even use the Internet connection to send information such as the images from a webcam, temperature sensor data.

B. SKETCHING IN HARDWARE

Artists, engineers, designers, architects, and makers frequently start their work with a simple sketch on paper. Putting the idea down as a sketch helps by pushing something from being abstract towards something more concrete, more real. Sketching something out also helps you communicate your idea to your peers and collaborators. But you don't necessarily need to use a paper and pencil to create a sketch. Having the power of a computer but the simplicity of Arduino development tools means that there's less to stand between you and your idea for an interactive object. It can help make the abstract idea a little more concrete. As a tool, the Galileo is meant to help you prototype early and iterate often so that you can get an idea of the look and feel of your design and rectify it without any inconvenience. I like to call this "sketching in hardware".

C. TOUR OF THE BOARD

First let's take a look at some of the important components on the Galileo which are shown in Figure: 2

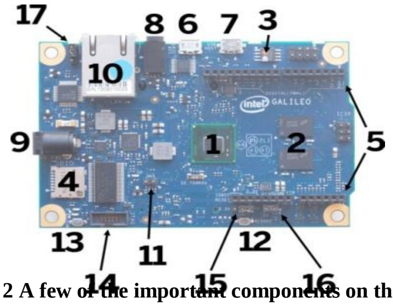


Fig. 2 A few of the important components on the Galileo

Processor (1)

The processor is the brains of the whole operation. Just like the central processing unit, or CPU, on your computer, it carries out all the instructions in your program by making calculations and reading or writing data in memory. This particular processor is Intel's Quark SoC X1000 Application Processor, which is designed for small-sized, low-power applications. It's not as powerful as your laptop's CPU, but it's much more powerful than the chip on an Arduino Uno.

Random-access memory (RAM) (2)

Random-access memory, or RAM, is where Galileo keeps running pro-grams and keeps track of data that's being used by those programs. The 10 Getting Started with Intel Galileo has 512 kilobytes of RAM built into the processor and an additional 256 megabytes of RAM on these chips. When the board is powered down, the data stored in RAM is lost. Any data that should be saved (such as your code) must be stored on the micro SD card.

Flash memory (3)

The flash memory acts like the hard drive of Galileo. Unlike with RAM, any data stored here is saved even after the board is shut down and the power is disconnected. For this reason, it's where the board's software and operating system are stored. It can hold 8 megabytes of data, most of which is taken up by Galileo's operating system.

Micro SD card slot (4)

If you need more space for larger programs or to store data, you can insert a micro SD card into this slot. Micro SD cards are just like the memory cards that you insert into a digital camera to save photographs. You can even load an operating system onto the card and boot off of it instead of the on-board Flash memory. You'll need to do this if you want additional functionality like Wi-Fi and access to the webcam because those drivers can't fit onto the 8 megabytes of on-board Flash memory. Galileo can use a card that's up to 32 gigabytes in size. See Appendix D for more information.

Arduino expansion pins (5)

Using these pins, you'll be able to connect to the inputs and outputs on the Galileo. You'll either use jumper wires to connect the pins to a bread-board for prototyping, or you'll use an Arduino shield to add functionality to your board.

USB client port (6)

You'll use this port to connect your Galileo to the USB port on your computer. Once it's connected, you can upload your code and communicate with it. Always connect the power supply before plugging the Galileo into your computer over USB.

USB host port (7)

This port allows you to connect USB computer peripherals to your Galileo. It could be accessories such as webcams, sound devices, storage, and much more.

Serial port (8)

This may look like a headphone jack, but it's not meant for audio. It's actually a serial port, used for interacting with the Galileo's Linux operating system through a text-based command line environment.

Power input (9)

This is where you'll plug in Galileo's power adapter. You must plug in the AC adapter when using Galileo. You must always power the board

Ethernet port (10)

The Ethernet port on the board will let you connect it to a wired network so that it can communicate with other computers and devices on the network, or access the Internet.

Mini PCI express slot (not pictured)

If you want to make your network connection wireless, you can connect a Wi-Fi card to the Mini PCI Express slot on the bottom of the board. This slot can also accommodate cards that offer additional functionality such as more storage space, GSM access for connecting to cellular networks, Bluetooth for wireless device connectivity, and much more.

Clock battery power (11)

This connector will let you wire up a small 3-volt coin cell battery to the Galileo so that the processor can keep track of the date and time even when the board is not connected to 5 volts.

Reboot button (12)

This button will reboot the board, including Galileo's Linux operating system.

Reset button (13)

This button will restart your code and send the reset signal to any shield attached to the expansion header. Galileo's Linux operating system will remain running as normal and won't restart.

JTAG header (14)

This 10-pin connector is mostly used by electrical engineers or advanced hobbyists to test and debug boards.

IOREF jumper (x3) (15)

This jumper lets you change the logic voltage level of the board from 5 volts to 3.3 volts for compatibility with 3.3-volt shields and components. Throughout this book, you'll be using Galileo in its 5-volt mode.

VIN jumper (16)

Pulling this jumper out will disconnect the VIN pin from Galileo's 5-volt regulator. If you're using a shield that requires more than 5 volts on the VIN pin, you must pull out this jumper to protect the board from damage.

I2C jumper (17)

This jumper lets you change the I2C address of a couple of on-board components. You may need to do this if you're using I2C components that conflict with the components on the board. In all likelihood, you won't need to use this.

D. GSM MODULE

GSM is digital cellular technology used for transmitting mobile voice and data services. GSM supports data transfer speeds of up to 9.6 Kbit/s and voice calls, together with the transmission of Short Message Service. GSM operates in 1.9GHz and 850MHz bands in the US and the 900MHz and 1.8GHz bands in Europe. The 850MHz band is also used for GSM and 3G in Canada, Australia and most of the South American countries. By having harmonized spectrum across most of the globe, GSM's international roaming capability allows users to access the same services when travelling abroad as at home. This gives consumers seamless and same number connectivity in more than 218 countries. GSM was originally defined for the 900 MHz range but after some time even the 1800 MHz range was used for cellular technology. The 1800 MHz range has its architecture and specifications almost same to that of the 900 MHz GSM technology but building the Mobile exchanges is easier and the high frequency Synergy effects add to the advantages of the 1800 MHz range.

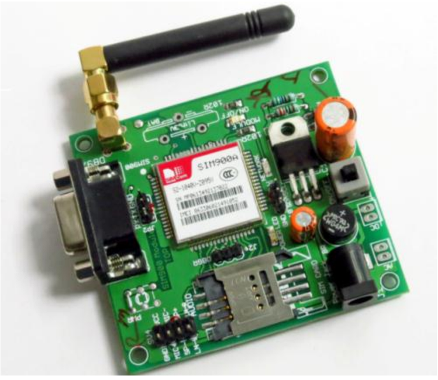


Fig. 3 A Top view of GSM module

New message indication +CNMI

Command syntax: AT+CNMI=<mode>,<mt>,<bm>,<ds>,<bfr>

Read message +CMGR

Command syntax: AT+CMGR=<index>

List message +CMGL

Command syntax: AT+CMGL=<stat>

Send message +CMGS

Command syntax in text mode:

AT+CMGS= <da> [,<toda>] <CR>

Text is entered <ctrl-Z / ESC >

E. LOAD CELL

A load cell is a transducer device that is used to create an electrical signal whose magnitude is directly proportional to the force being measured. The various types of load cells include hydraulic load cells, pneumatic load cells and strain gauge load cells.

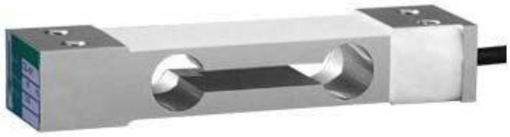


Fig. 4 Load Cell

Through a mechanical construction, the force being sensed deforms a strain gauge. The strain gauge measures the contortion as a change in resistance, which is a measure of the applied forces. A load cell usually consists of four strain gauges in a Wheatstone bridge configuration. Load cells of Quarter Bridge or half bridge are also available. The electrical signal output is typically in the order of a few milli volts and needs amplification with the help of an instrumentation amplifier before it can be further processed. The output of the load cell can be scaled to determine the force applied to the load cell.

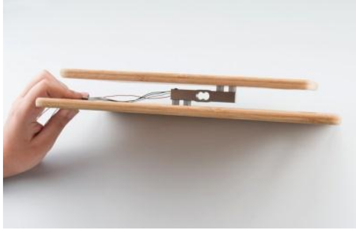


Figure: 5 Load Cell Setup

Strain gauge load cells convert the load acting on them into electrical signals. The gauges themselves are affirmed onto a beam or structural member that contorts when weight is applied. To obtain maximum sensitivity and temperature compensation four strain gauges are used. Two of them are in tension, and two in compression, and are wired with compensation adjustments. The strain gauge load cell is fundamentally a spring developed for strain measurement. Gauges are fastened in areas that exhibit strain in compression or tension. The gauges are mounted in a differential bridge to enhance measurement accuracy. The strain changes the electrical resistance of the gauges in proportion to the weight applied. Other load cells are fading into obscurity, as strain gauge load cells have increased accuracy levels and affordable unit costs.

Nonlinearity: The load cells tend to be nonlinear at the low end of their scale. This becomes important for cells with large surplus of load capability or sensing very large ranges, or to withstand temporary overloads or shocks like the rope clamps. So there is a need for more points for the calibration curve.

Wiring: The full-bridge cells are generally available in four-wire configuration. The wires to the sides are the signal (labeled S+ and S−), the wires to the top and bottom end of the bridge are the excitation (often labeled E+ and E−, or Ex+ and Ex−). Ideally, the voltage difference between S+ and S− is zero under zero loads, and grows proportionally to the load cell's mechanical load.

Wiring colors: The most common color assignment is black for Ex−, red for Ex+, white for S− and green for S+. Less common assignments are white for Ex−, red for Ex+, blue for S− and green for S+, or red for Ex+, blue for Ex−, green for S+, and yellow for S−.

III. INA125 INSTRUMENTATION AMPLIFIER

The INA125 is a high accuracy, low power instrumentation amplifier with a precision voltage reference. It provides precision differential-input amplification and complete bridge excitation on a single integrated circuit. A single external resistor chooses any gain in the range of 4 -10,000. The INA125 is laser-trimmed for low offset voltage (250µV), high common-mode rejection (100dB at G = 100) and low offset drift (2µV/°C) . It is capable of operating on either a single (+2.7V to +36V) or dual (±1.35V to ±18V) supplies. The reference voltage is accurate to ±0.5% (max) with ±35ppm/°C drifts (max). Sleep mode allows shutdown and duty cycle operation to save power. The INA125 is available in 16-pin plastic DIP and SO-16 surface-mount packages and is specified for the –40°C to +85°C industrial temperature range.

Fig. 6 INA125P IC

IV.HARDWARE RESULTS

It was designed such that the load value from the load cell decreases from a pre determined value given in the code, making it easier to book the refill.

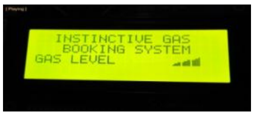


Fig. 7 LCD Display showing Weight of Gas Cylinder

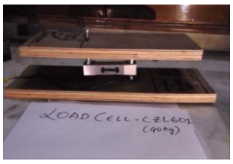
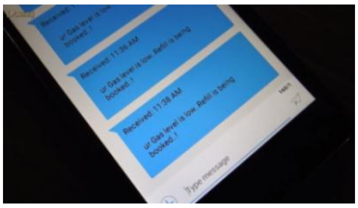


Fig.8 Load Cell Setup

Fig 9 Total System Setup



Received:

ur Gas level is low. Refill is being booked. !

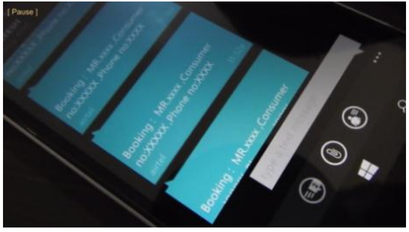
Received: 11:36 AM

ur Gas level is low. Refill is being booked. !

Received: 11:38 AM

ur Gas level is low. Refill is being booked. !

Fig.10 Message in Consumer mobile Number



Booking : MR.xxxx ,Consumer no:XXXXX ,Phone no:XXXX

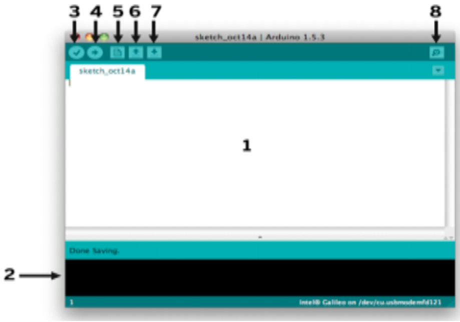
Booking : MR.xxxx ,Consumer no:XXXXX ,Phone no:XXXX

Booking : MR.xxxx ,Consumer

Fig.11 Message at LPG Provider

V.SOFTWARE

Arduino is an open-source prototyping platform based on easy-to-use hardware and software. The programming is done with Arduino programming language (based on Wiring), and the Arduino Software (IDE), based on processing. The Arduino Software (IDE) runs on Windows, Macintosh OSX, and Linux operating systems.When you open the IDE for the first time, you'll be presented with a window for a new sketch. A sketch is an Arduino project's code files. Here are a few of the parts of the IDE.


3
4
5
6
7
8
sketch_oct14a | Arduino 1.5.3
sketch_oct14a
1
2
1
Intel® Galileo on /dev/cu.usbmodemfd121

Done Saving.

Fig. 12 Basic Arduino Sketch Area with all options

Code entry area (1) this is where you'll type in the code for your sketch.

Console (2) this is where you'll see status messages when uploading your code to the board. Verify

Button (3) Click this button to verify that your code has no errors. If your code does have errors, they will be displayed in the console.

Upload button (4) Click this button to verify the code and upload it to the board if there are no errors.

New button (5) this creates a new sketch in the current window.

Open button (6) this will allow you to open a saved sketch from your Sketch book folder, the location on your computer where the Arduino IDE saves your sketches.

Save button (7) this will allow you to save the sketch in the current window. You'll be prompted to enter a file name if you haven't already saved the file.

Serial monitor (8) this button opens the serial monitor, which lets you send and receive Information between your board and your computer.

Connecting the Board The next step is to power on the Galileo and connects it to your computer so that you can upload your code to the board.

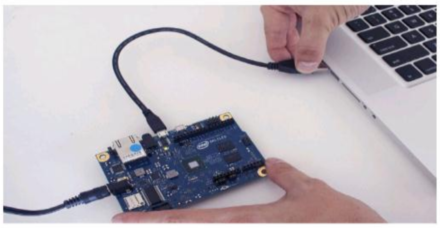


Fig. 13 Connecting Intel Galileo Gen2 to PC

VI.CONCLUSION AND FUTURE WORK

In many circumstances this system proves useful to the people living in rural areas, Villages and also to the lower and middle class societies in cities. The design is mainly intended to monitor the level of remaining fuel in the cylinder. It can be extended by introducing Safety Mechanism, which sends alert intimation to the respective authorities (say Fire station, Consumer) when there is a leakage in the gas or when there is an emergency fire occurring as a result of this leakage.

ACKNOWLEDGEMENT

I would like to take this opportunity to express our profound sense of gratitude to **CMR Technical Campus** for their motivation and encouragement all the way during this Work.

REFERENCES

[1] B. Koo, "Washing machine water level detecting method using weight sensor," U.S. Patent 5 720 064, 1998.

[2] Z. Wang, H. Hu, "Analysis and Optimization of a Compliant Mechanism-Based Digital Force/Weight Sensor," IREE Sensors, vol. 5, no. 6, pp. 1243-1249, Dec. 2005.

[3] H. G. Rodney Tan, C. H. Lee and V. H. Mok, "Automatic Power Meter Reading System Using GSM Network," in Proc. of the 8th International Conference (IPEC2007), pp. 465-469, 2007.

[4] M. Yuchun, H. Yinghong, Zhang Kun and Li Zhuang, "General Application Research on GSM Module," IEEE 8th International Conference on Internet Computing and Information Services (IPEC), 2007.

[5] Y. Mengda and Zhu Min, "A Greenhouse Remote Monitoring System Based on GSM," in Proc. of IEEE International Conference on Information Management, Innovation Management and Industrial Engineering, 2011.

AUTHORS

TANNERU NARENDRA KUMAR received his B.Tech degree in Electronics and Communication Engineering in 2012 and M.Tech Degree in Embedded Systems in 2014.Presently working as an Assistant Professor in CMR Technical Campus (ECE Dept) Hyderabad. His Area of interest is Automotive Embedded Sytems.

M.SRAVANTHI received her B.Tech Degree in Electronics and Communication Engineering in 2007 and M.Tech Degree in Embedded Systems in 2011.Presently she is working as an Assistant professor in CMR Technical Campus (ECE Dept) Hyderabad. Her area of interest is in the field of Embedded Systems.

SK.DILSHAD received her B.Tech Degree in Electronics and Communication Engineering in 2009 and M.Tech Degree in VLSI Design in 2012.Presently she is working as an Assistant professor in CMR Technical Campus (ECE Dept) Hyderabad. Her area of interests is in the fields of VLSI System Design and Embedded Systems.