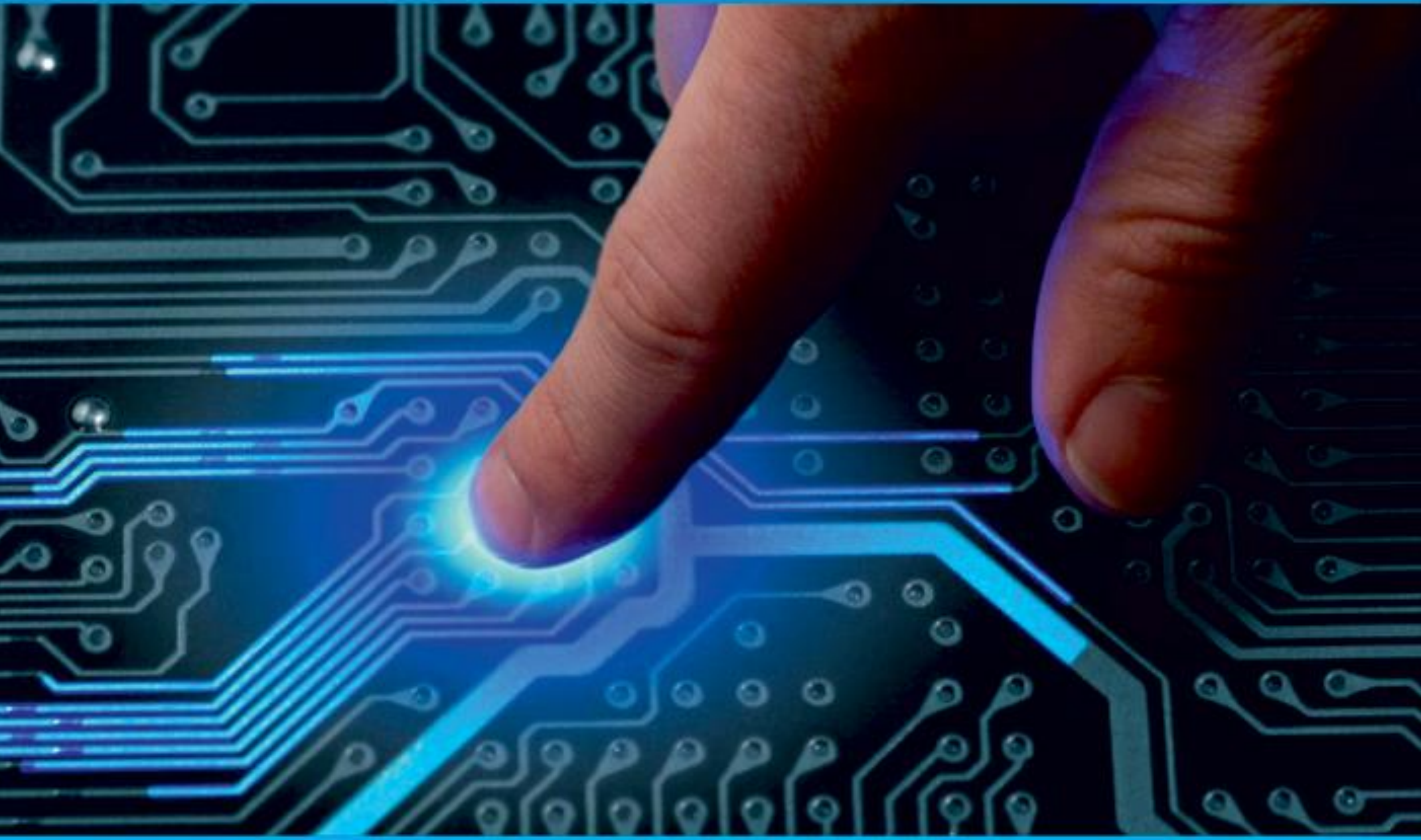
IJIRCCE

e-ISSN: 2320-9801 | p-ISSN: 2320-9798



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

Volume 12, Issue 4, April 2024

INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA

Impact Factor: 8.379

International Journal of Innovative Research in Computer and Communication Engineering

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 4, April 2024 ||

| DOI: 10.15680/IJIRCCE.2024.1204351 |

IoT Based Smart Parking System

Mrs. R. Prathipa[1]., Mulam Reddy Hemanth Krishna Reddy[2], Sharif Mohammad Shahid[3] And Bodireddy Vinod Kumar Reddy[4]

Assistant Professor, Department of Electronics and Communication Engineering, Panimalar Institute of Technology, Chennai, India[1].

Student, Department of Electronics and Communication Engineering, Panimalar Institute of Technology, Chennai, India[234]

ABSTRACT: The main objective is to design this project is to overcome the vehicle parking issues. Now a day we can see that traffic is increasing day by day and in weekend days when we go to cinema halls, malls, parks we are facing much issues while parking our vehicle. And due to this our time and fuel is also wasted. So, to overcome these issues, we are implementing this project. In this project there are also some important features will be available such as identification of each vehicle, display availability of parking slots and many more.

INDEX TERMS Ultrasonic ranging sensor, ESP8266 WiFi Module, Parking Sensors, Processing Unit, Mobile Application, The Cloud, Some Cameras

I. INTRODUCTION

Today, the parking industry is being transformed by new technologies that are allowing cities to reduce rates of congestion significantly. Sensor networks that sense vehicle occupancy are providing the basic intelligence behind smart parking systems. Thanks to the Smart Parking technology, it is now possible to know in real-time the location of free parking spaces and to help drivers to get to their ultimate destination. The wireless sensors are still intrusive; they are embedded in the pavement, or taped to the surface of each individual parking lot.

There was a requirement of separate person to manage the whole parking system i.e. to check whether the parking slot is free or not, allotment of parking slot for new incoming vehicle. As these activities were done manually, it was so time consuming and there were also no facilities of recording the vehicle number. Some of the customers have to wait over long time to get the parking slot.

A simple and easy task such as parking is thought as a tedious and time-consuming process due to mismanagement of parking system.

II. LITERATURE REVIEW

1. AN IOT-BASED INTELLIGENT SYSTEM FOR REAL-TIME PARKING MONITORING AND AUTOMATIC BILLING

This paper describes an internet of things (IoT)-based parking sensing system that deploys a robust outdoor vehicle localization and recognition methodologies
AUTHOR: Riad Kanan; Houssam Arbess

2. SMART PARKING SYSTEM USING IOT TECHNOLOGY

This paper presents a Smart Parking & Energy Management solution for a structured environment such as a multi-storied office parking area. The system proposes implementation of state-of-the-art Internet of Things (IoT) technology to mold with advanced Honeywell sensors and controllers to obtain a systematic parking system for users.

AUTHOR: Akshat Tiwari, Vipul Jirge.

3. IOT BASED AUTOMATED PAID PARKING USING ELECTROMAGNETIC RFID TAG

This paper comes with building an efficient system that performs the task of identifying free slots in a parking area and maintaining the time and paying duration and also to have record about the parked vehicles. This system also does the job of calculating the payment and collecting thereby reducing the human efforts and time wasted on line to a great extent.

AUTHOR: P.M. Akshay; K. Murugesh; Yashketan Patra

III. METHODOLOGY FLOW

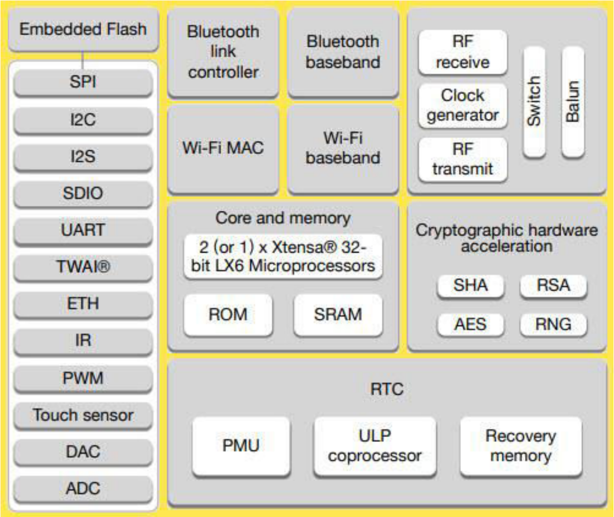
3.1 Arduino:

Arduino is a great platform for beginners into the World of Microcontrollers and Embedded Systems. With a lot of cheap sensors and modules, you can make several projects either as a hobby or even commercial.As technology advanced, new project ideas and implementations came into play and one particular concept is the Internet of Things or IoT. It is a connected platform, where several “things” or devices are connected over internet for exchange of information.In DIY community, the IOT projects are mainly focused on Home Automation and Smart Home applications but commercial and industrial IoT projects have far complex implementations like Machine Learning, Artificial Intelligence, Wireless Sensor Networks etc.

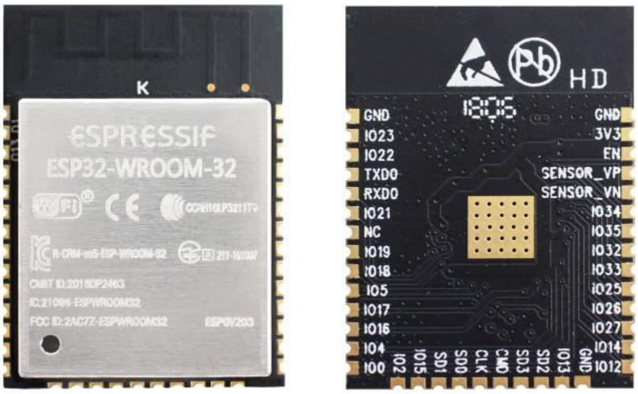
3.2 ESP32:

ESP32 is a low-cost System on Chip (SoC) Microcontroller from Espressif Systems, the developers of the famous ESP8266 SoC. It is a successor to ESP8266 SoC and comes in both single-core and dual-core variations of the Tensilica’s 32-bit Xtensa LX6 Microprocessor with integrated Wi-Fi and Bluetooth.

The good thing about ESP32, like ESP8266 is its integrated RF components like Power Amplifier, Low-Noise Receive Amplifier, Antenna Switch, Filters and RF Balun. This makes designing hardware around ESP32 very easy as you require very few external components.



Another important thing to know about ESP32 is that it is manufactured using TSMC’s ultra-low-power 40 nm technology. So, designing battery operated applications like wearables, audio equipment, baby monitors, smart watches, etc., using ESP32 should be very easy.



3.3 POWER SUPPLY:
ADAPTER:

GENERAL DESCRIPTION

An adapter is a device that converts attributes of one electrical device or system to those of an otherwise incompatible device or system. Some modify power or signal attributes, while others merely adapt the physical form of one electrical connector to another. In a computer, an adapter is often built into a card that can be inserted into a slot on the computer's motherboard. The card adapts information that is exchanged between the computer's microprocessor and the devices that the card supports.

PRODUCT DESCRIPTION

An electric power adapter may enable connection of a power plug, sometimes called, used in one region to a AC power socket used in another, by offering connections for the disparate contact arrangements, while not changing the voltage. An AC adapter, also called a "recharger", is a small power supply that changes household electric current from distribution voltage) to low voltage DC suitable for consumer electronics. Some modify power or signal attributes, while others merely adapt the physical form of one electrical connector to another. For computers and related items, one kind of serial port adapter enables connections between 25-contact and nine-contact connectors, but does not affect electrical power- and signalling-related attributes.

3.4 IR SENSOR

GENERAL DESCRIPTION

IR LED emits infrared radiation. This radiation illuminates the surface in front of LED. Depending on reflectivity of the surface, amount of light reflected varies. This reflected light is made incident on reverse biased IR sensor. The amount of electron-hole pairs generated depends on intensity of incident IR radiation. Thus, as intensity of incident ray varies, voltage across resistor will vary accordingly.

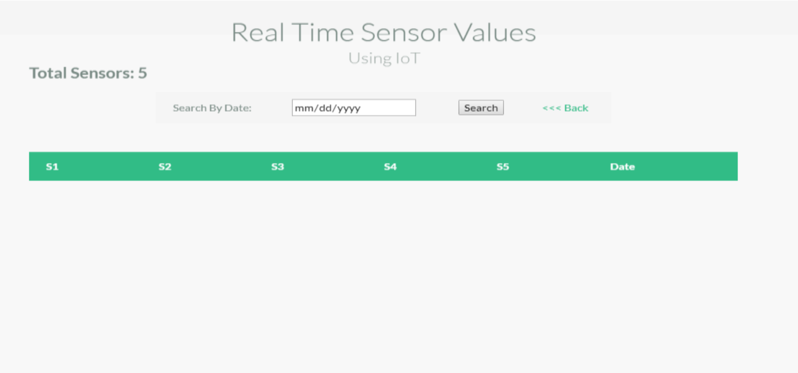
PRODUCT DESCRIPTION

An infrared sensor is an electronic device, that emits in order to sense some aspects of the surroundings. An IR sensor can measure the heat of an object as well as detects the motion. Usually in the infrared spectrum, all the objects radiate some form of thermal radiations. These types of radiations are invisible to our eyes, that can be detected by an infrared sensor. The emitter is simply an IR LED (Light Emitting Diode) and the detector is simply an IR photodiode which is sensitive to IR light of the same wavelength as that emitted by the IR LED. When IR light falls on the photodiode, The resistances and these output voltages, change in proportion to the magnitude of the IR light received.

3.5 IOT

The Internet of things (IoT) is the network of everyday objects — physical things embedded with electronics, software, sensors, and connectivity enabling data exchange. Basically, a little networked computer is attached to a thing, allowing information exchange to and from that thing. Be it lightbulbs, toasters, refrigerators, flower pots, watches, fans, planes, trains, automobiles, or anything else around you, a little networked computer can be combined with it to accept input (especially object control) or to gather and generate informational output (typically object status or other sensory data).
This means computers will be permeating everything around us — ubiquitous embedded computing devices, uniquely identifiable, interconnected across the Internet. Because of low-cost, networkable microcontroller modules, the Internet of things is really starting to take off.

3.5.1 WEB SERVER



Real Time Sensor Values

Using IoT

Total Sensors: 5

Search By Date: mm/dd/yyyy Search <<< Back

S1	S2	S3	S4	S5	Date

Espresso's ESP32EX delivers highly integrated Wi-Fi SoC solution to meet users' continuous demands for efficient power usage, compact design and reliable performance in the Internet of Things industry.

3.5.2 Channel Frequencies:

The RF transceiver supports the following channels according to IEEE802.11b/g/n standards

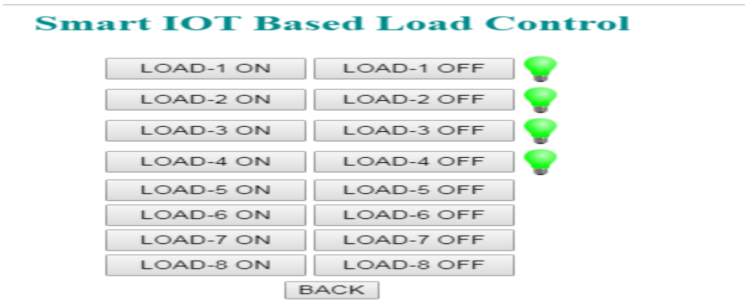
GHz Receiver:

The 2.4 GHz receiver down-converts the RF signals to quadrature baseband signals and converts them to the digital domain with 2 high-resolution high-speed ADCs. To adapt to varying signal channel conditions, RF filters, automatic gain control (AGC), DC offset cancelation circuits and baseband filters are integrated within ESP32EX.

GHz Transmitter:

The 2.4 GHz transmitter up-converts the quadrature baseband signals to 2.4 GHz, and drives the antenna with a high-power CMOS power amplifier. The function of digital calibration further improves the linearity of the power amplifier, enabling a state of art performance of delivering +19.5 dBm average power for 802.11b transmission and +16 dBm for 802.11n transmission.

3.5.3 WEB SERVER: Controlling Section



FEATURES

- Power Supply: DC +12v 1Amp.
- Auto data updating: 30sec
- Digital Output port Pins: +5V DC
- Message Format: *message or Data # (Start with * and End with #)
- Provided with 3 links
- Data updating to a specific web site
- Device controlling web site
- Data updating to a social network

APPLICATIONS

- Online Traffic monitoring
- Online Health monitoring
- Real time Transport and Logistics monitoring
- Daily life and domestics

3.6 L293D DC MOTOR DRIVER MODULE

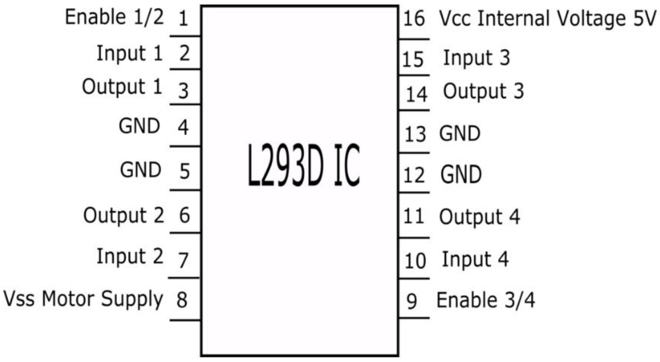
The project designed around L293D IC. The L293D device is quadruple high-current half-H driver. The 293D is designed to provide bidirectional drive current up to 600mA a voltage from 5V to 36V. L293D Adapter Board can be used as dual DC motor driver or bipolar stepper motor driver. Useful in robotics application, bidirectional DC motor controller and stepper motor driver. Separate logic supply to reduce dissipation. L293D includes the output clamping diodes for protections.

SPECIFICATIONS

- Motor/Logic supply 5 to 36 V
- Logic controls input 7 VDC max
- Inhibit facility/enable
- High Noise immunity

L293D IC Pin Out

The L293D is a 16 pin IC, with eight pins, on each side, to controlling of two DC motor simultaneously. There are 4 INPUT pins, 4 OUTPUT pins and 2 ENABLE pin for each motor.



DC MOTOR

GENERAL DESCRIPTION

The relationship between torque vs speed and current is linear as shown left; as the load on a motor increases, Speed will decrease. The graph pictured here represents the characteristics of a typical motor. As long as the motor is used in the area of high efficiency (as represented by the shaded area) long life and good performance can be expected.

PRODUCT DESCRIPTION

Geared dc motors can be defined as an extension of dc motors. A geared DC Motor has a gear assembly attached to the motor. The speed of motor is counted in terms of rotations of the shaft per minute and is termed as RPM .The gear assembly helps in increasing the torque and reducing the speed. Using the correct combination of gears in a gear motor, its speed can be reduced to any desirable figure. This concept where gears reduce the speed of the vehicle but increase its torque is known as gear reduction. A DC motor can be used at a voltage lower than the rated voltage. But, below 1000 rpm, the speed becomes unstable, and the motor will not run smoothly.

3.7 RFID READER

GENERAL DESCRIPTION

A Radio Frequency Identification Reader (RFID reader) is a device used to gather information from an RFID tag, which is used to track individual objects. Radio Frequency waves are used to transfer data from the tag to a reader. The RFID tag it must be within the range of an RFID reader, in order to be read. RFID technology allows several items to be quickly scanned and enables fast identification of a particular product, even when it is surrounded by several other items.

PRODUCT DESCRIPTION

Radio frequency identification (RFID) is one method for Automatic Identification and Data Capture (AIDC). RFID tags are used in many industries. An RFID system consists of three components: an antenna and transceiver and a transponder. The antenna uses radio frequency waves to transmit a signal that activates the transponder. When activated, the tag transmits data back to the antenna. An RFID reader's function is to interrogate RFID tags. The means of interrogation is wireless and because the distance is relatively short; line of sight between the reader and tags is not necessary. A reader contains an RF module, which acts as both a transmitter and receiver of radio frequency signals.

RFID Reader

IV. NAMED REGISTERS

Embedded C allows direct access to processor registers that are not addressable in any of the machine's address spaces. The processor registers are defined by the compiler-specific, named-register, storage class for each supported processor. The processor registers are declared and used like conventional C variables (in many cases volatile variables). Developers using Embedded C can now develop their applications, including direct access to the condition code register and other processor-specific status flags, in a high-level language, instead of inline assembly code.

Named address spaces and full processor access reduces application dependency on assembly code and shifts the responsibility for computing data types, array and structure offsets, and all those things that C compilers routinely and easily do from developers to compilers.

I/O HARDWARE ADDRESSING

The motivation to include primitives for I/O hardware addressing in Embedded C is to improve the portability of device-driver code. In principle, a hardware device driver should only be concerned with the device itself. The driver operates on the device through device registers, which are device specific. However, the method to access these registers can be very different on different systems, even though it is the same device that is connected. The I/O hardware access primitives aim to create a layer that abstracts the system-specific access method from the device that is accessed. The ultimate goal is to allow source-code portability of device drivers between different systems. In the design of the I/O hardware-addressing interface, three requirements needed to be fulfilled:

1. The device-drive source code must be portable.
2. The interface must not prevent implementations from producing machine code that is as efficient as other methods.
3. The design should permit encapsulation of the system-dependent access method.

The design is based on a small collection of functions that are specified in the <iohw.h> include file. These interfaces are divided into two groups; one group provides access to the device, and the second group maintains the access method abstraction itself.

To access the device, the following functions are defined by Embedded C:

```
unsigned int iord(  ioreg_designator );
void iowr(  ioreg_designator, unsigned int value );
void ioor(  ioreg_designator, unsigned int value );
void ioand( ioreg_designator, unsigned int value );
void ioxor( ioreg_designator, unsigned int value );
```

These interfaces provide read/write access to device registers, as well as typical methods for setting/resetting individual bits. Variants of these functions are defined (with buf appended to the names) to access arrays of registers. Variants are also defined (with l appended) to operate with long values.

All of these interfaces take an I/O register designator ioreg_designator as one of the arguments. These register designators are an abstraction of the real registers provided by the system implementation and hide the access method from the driver source code. Three functions are defined for managing the I/O register designators. Although these are abstract entities for the device driver, the driver does have the obligation to initialize and release the access methods. These functions do not access or initialize the device itself because that is the task of the driver. They allow, for example, the operating system to provide a memory mapping of the device in the user address space.

```
void iogroup_acquire( iogrp_designator );
void iogroup_release( iogrp_designator );
void iogroup_map( iogrp_designator, iogrp_designator );
```

The iogrp_designator specifies a logical group of I/O register designators; typically this will be all the registers of one device. Like the I/O register designator, the I/O group designator is an identifier or macro that is provided by the system implementation.The map variant allows cloning of an access method when one device driver is to be used to access multiple identical devices.

EMBEDDED C PORTABILITY

By design, a number of properties in Embedded C are left implementation defined. This implies that the portability of Embedded C programs is not always guaranteed. Embedded C provides access to the performance features of DSPs. As not all processors are equal, not all Embedded C implementations can be equal For example, suppose an application requires 24-bit fixed-point arithmetic and an Embedded C implementation provides only 16 bits because that is the native size of the processor. When the algorithm is expressed in Embedded C, it will not produce outputs of the right precision.

In such a case, there is a mismatch between the requirements of the application and the capabilities of the processor. Under no circumstances, including the use of assembly, will the algorithm run efficiently on such a processor. Embedded C cannot overcome such discrepancies. Yet, Embedded C provides a great improvement in the portability and software engineering of embedded applications. Despite many differences between performance-specific processors, there is a remarkable similarity in the special-purpose features that they provide to speed up applications.

Writing C code with the low-level processor-specific support may at first appear to have many of the portability problems usually associated with assembly code. In the limited experience with porting applications that use Embedded C extensions, an automotive engine controller application (about 8000 lines of source) was ported from the eTPU, a 24-bit special-purpose processor, to a general-purpose 8-bit Freescale 68S08 with about a screen full of definitions put into a single header file. The porting process was much easier than expected. For example, variables that had been implemented on the processor registers were ported to unqualified memory in the general-purpose microprocessor by changing the definitions in the header definition and without any actual code modifications. The exercise was to identify the porting issues and it is clear that the performance of the special-purpose processor is significantly higher than the general-purpose target.

V. CONCLUSION

A novel intelligent parking sensor system was presented. It allows real time parking monitoring along with parking payment without requiring any user/driver interaction. Thanks to the proposed innovative approach, the sensor system has advantages in terms of detection and payment reliability and reduces costs by reducing system complexity, investing in infrastructure and replacing batteries.

REFERENCES

1. Z. Zhang, X. Li, H. Yuan, and F.Yu, “A Street Parking System Using Wireless Sensor Networks,” International Journal of Distributed Sensor Networks, 2013.
2. Q.-J. Kong, Z. Li, Y. Chen, and Y. Liu, “An approach to Urban traffic state estimation by fusing multisource information,” IEEE Transactions on Intelligent Transportation Systems, vol. 10, no. 3, pp. 499–511, 2009.
3. G. Alessandretti, A. Broggi, and P. Cerri, “Vehicle and guard rail detection using radar and vision data fusion,” IEEE Transactions on Intelligent Transportation Systems, vol. 8, no. 1, pp. 95–105, 2007.

| DOI: 10.15680/IJIRCCE.2024.1204351 |

4. P. N. Pathirana, A. E. K. Lim, A. V. Savkin, and P. D. Hodgson,"Robust video/ultrasonic fusion-based estimation for automotive applications," IEEE Transactions on Vehicular Technology, vol. 56, no. 4, pp. 1631– 1639, 2007.
5. H. Bura, et. al, "An Edge Based Smart Parking Solution Using Camera Networks and Deep Learning," 2018 IEEE International Conference on Cognitive Computing (ICCC), pp. 17-24, 2018.
6. Q. Wu, C. Huang, S. Y. Wang, W. C. Chiu, and T. Chen, "Robust Parking Space Detection Considering Inter-Space Correlation," IEEE international Conference on Multimedia and Expo, Beijing, China, pp. 659-662, 2007.
7. L. Zhang, R. Wang, and L. Cui, "Real-time traffic monitoring with magnetic sensor networks," Journal of Information Science and Engineering, vol. 27, no. 4, pp. 1473–1486, 2011.
8. J. Zhu, H. Cao, J. Shen, and H. Liu, "Data fusion for magnetic sensor based on fuzzy logic theory," in Proceedings of the 4th International Conference on Intelligent Computation Technology and Automation (ICICTA '11), pp. 87–92, March 2011.
9. C. Trigona et al., "Implementation and characterization of a smart parking system based on 3-axis magnetic sensors," 2016 IEEE International Instrumentation and Measurement Technology Conference Proceedings, Taipei, 2016, pp. 1-6.
10. B. Yang and Y. Lei, "Vehicle detection and classification for low speed congested traffic with anisotropic magnetoresistive sensor," IEEE Sensors J., vol. 15, no. 2, pp. 1132–1138, Feb. 2015
11. Z. Zhang, M. Tao, and H. Yuan, "A parking occupancy detection algorithm based on AMR sensor," IEEE Sensors J., vol. 15, no. 2, pp. 1261–1269, Feb. 2015.
12. R. Kanan, A. Sweleh, and O. Elhassan, "A Self-powered and Wireless Parking Sensor with Lux-RSS Correlation for Extra Accuracy," 2019 International Conference on Electrical and Computing Technologies and Applications (ICECTA), 2019.
13. B. Cai, R. Alvarez, M. Sit, F. Duarte, and C. Ratti, "Deep LearningBased Video System for Accurate and Real-Time Parking Measurement," IEEE Internet of Things J., vol. 6, no. 5, pp. 7693–7701, Oct. 2019.
14. Transportation Cost and Benefit Analysis II: Parking Costs, VTP Inst., Victoria, BC, Canada, 2018.
15. SMART, "Vehicle parking," United States of America Patent 0142534A1, 2015.

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

☎ 9940 572 462 6381 907 438 ijircce@gmail.com

www.ijircce.com

Scan to save the contact details