MICROCONTROLLER BASED DATALOGGER FOR TRAFFIC INFORMATION COLLECTION IN PARKING AREAS

Luciano Spinello[1], **Luca Spaghetti**[1], **Francesco Martinelli**[1], **Valentino Di Salvo**[2]

[1] *Dipartimento di Informatica Sistemi e Produzione*
Università di Roma Tor Vergata
I-00133 Rome, Italy
e-mail: `lucianospinello@universitor.it`,
`lucaspaghetti@fastwebnet.it`,
`martinelli@disp.uniroma2.it`

[2] *Traffic Engineer*
e-mail: `valentinoambrogio.disalvo@fastwebnet.it`

Abstract: In this paper an automatic device to collect data regarding open parking areas is described. It is based on a loop vehicle detector and a datalogger where collected data are recorded for subsequent processing operations on a standard PC. The device is controlled through a commercial microcontroller unit. The circuit scheme and firmware design are discussed in detail in the paper. Experimental results are provided in the paper together with a discussion of related realizations available in the technical literature.

Keywords: Traffic Data Collection, Microprocessor control, Data logging, Sensors.

1. INTRODUCTION

For statistical purposes, parking companies regularly collect data regarding the occupancy state of public parking areas. Each park slot should be monitored on a basis of $24h$ per day. This kind of study is usually realized by a manual procedure, which is very expensive and subject to many errors.

For this reason, it is of interest to design and realize a device to be placed in each park slot to be monitored, performing the human work in a more precise and efficient way.

However, the requirements specified for this device are very restrictive:

- it should be easy to place it in a standard car parking slot;
- it should have an autonomy of 7 days (night and day) using a common battery;
- it should be sufficiently robust to environment disturbances;
- it should be realized with commercial and low cost components.

At present, such a kind of device is not available on the market. A large part of traffic data collection devices have been realized to perform flow measurements, especially in a situation of heavy traffic conditions. It would be very hard to adapt

[1] This work has been partially supported by ASI and MIUR (cofin: MATRICS).

these devices to our purposes, because the requirements are completely different. If a traffic data collection device accidentally skips the passage of a few vehicles, the relative error committed is usually negligible; in fact this kind of devices is used on highways where the typical traffic flow ranges in the order of 2500 and more vehicles per hour. On the contrary, the error could be dramatic if the requested park slot device skips a vehicle which takes busy the park slot for several days. In addition, a parking measurement device is a finite state machine, unlike a flow traffic counter, since operations performed depend on the park slot state (free or busy).

A problem similar to the one addressed in this paper, even if in a domain completely different, is considered in Luharuka and Krishnamurti (2002). The objective in Luharuka and Krishnamurti (2002) is the realization of a portable datalogger for physiological sensing. The requirements of this device are an autonomy of 12h and an high reliability. The projected device comprises a commercial GSR (galvanic skin response) sensor which is able to provide output data like heart rate, temperature and blood pressure. An onboard datalogger containing a great amount of eeprom guarantees the recording of 7-20 MB data corresponding to 12 hours logging report. Such data can be downloaded on a PC at the end of a recording session.

Many similarities can be observed between the device described in this paper and the one presented in Luharuka and Krishnamurti (2002). They are based on the same control unit family microprocessor (MPU Microchip PIC16XXXX), a very high quality and low cost component; they both use a battery as unique power supply, with voltages conversions where needed.

It would be really appealing (and it is actually considered as a future development of our project) the possibility of realizing power independent measurement devices connected to a central data collection system, like the traffic information collection system realized by Koito industries and described in Matsuda (1999). In the system described in Matsuda (1999), detectors are powered by small solar cells and communicate detected data to a central device through free-space optical transmission links. Vehicle detectors are based on ultrasonic sensors. This system was installed in February 1999 along the national route 246 to collect traffic information regarding vehicles running east and west between Tokyo and Shizuoka. In this area a commercial power supply is not available.

The paper is organized as follows: in Section 2 we discuss the hardware design, in Section 3 we describe the control algorithm and its implementation, in Section 4 we present some experimental results, finally, in Section 5, we sketch some concluding remarks.

2. HARDWARE DESIGN

2.1 A sensor survey.

The first step to realize the requested device is the selection of a sensor or a proper combination of sensors with high reliability, acceptable costs and low energy consumption. The following sensors have been considered: inductive/capacitive proximity switches, photoelectric sensors, microwave sensors, vibration based monitoring sensors, ultrasonic proximity switches, loop vehicle detectors. Videocameras have not been considered mainly for privacy violation reasons.

For different reasons, the loop vehicle detector has appeared as the most promising sensor. It is characterized by medium energy consumption (65mA), an optimal sensing range and an adjustable sensibility. This kind of sensor is commonly used in automatic switched gate parking areas to detect the presence of vehicles which approach to an exit gate. The sensor is small for dimension and weight but the loop requires a surface of about 1 m^2.

2.2 Model concept

A long time has been spent to understand which sensors and which components should have been considered and in which way they should have been disposed into the parking slot. At the end of these investigations, the *Inductive model* has been selected, which is based on a loop vehicle detector sensor, located in a parking slot as illustrated in figure 1.

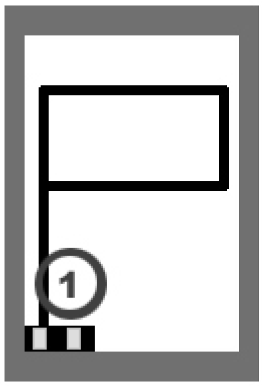

Fig. 1. Sensor placement in a park slot (inductive model)

Data are recorded in a HoboEvent Datalogger produced by the OnSet Computers. The overall device is controlled by a MPU Microchip PIC16F84, which operates as a two state finite machine, one state corresponding to *Empty parking* and the other corresponding to *Busy parking*. Energy is given by a unique commercial 12V battery.

The control device, sensor and datalogger units, are located inside a rigid box placed near the sidewalk, while the loop of the vehicle detector has to be located in the middle of the parking slot under a special PVC covering. The vehicle detector sensor is always on during the overall measurement session. The control algorithm works according to the following logic: when a vehicle enters a park slot, its presence is immediately detected but the entrance event is registered in the datalogger only if the vehicle stays in the park slot for at least 20 s, as explained below.

2.3 The FG1 detector

In this subsection we provide a detailed description of the sensor used in the realized device, the FG1 vehicle loop detector produced by FAAC (see Fig. 2). It is commonly used for gate automation systems and produced for this kind of applications.

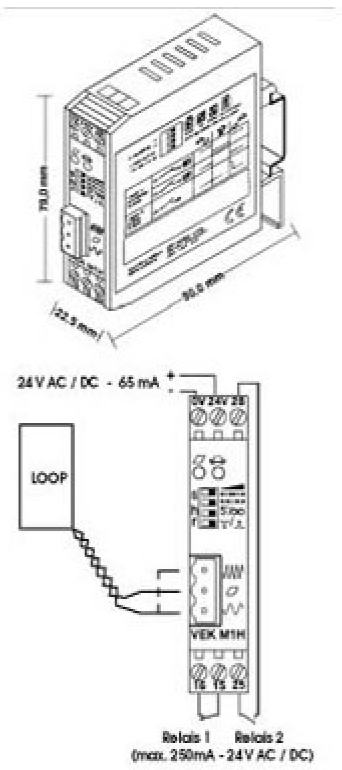

Fig. 2. Detector FG1

The operation of the FG1 Detector is based on a Hartley type HF-circuit, where the oscillator inductor is the loop. When a metallic body (like a car) is over the loop, the inductance and consequently the circuit natural oscillation frequency are changed. A PLL discriminator transforms this frequency variation in a digital signal. By properly tuning some adjustable parameters, it is possible to have precise measurements, ignoring frequency variations caused by environment disturbances. The FG1 Detector would require a 24V power supply but it correctly works using a 12V battery.

Two different outputs, corresponding to two different electro-mechanical relays, give respectively the signal of *busy loop* and of *loop in exit transition*.

Different parameters can be set to obtain the desired behavior: the vehicle loop detector sensitivity, the circuit frequency oscillation and the operation principles of the output relays.

The vehicle loop detector sensitivity is a really important parameter to be selected. The precision of the measurements depends on its good tuning and it has been set only after a large campaign of experiments. The circuit frequency oscillation characterizes the velocity of the sensor of detecting a state transition. For parking purposes a low setting of this parameter is acceptable and allows to save energy. The operation principles of the output relays can be set in two ways. In the first operation way, if the loop is busy, the corresponding relays is closed, while it would remain open in the second operation way. The other relay (the one corresponding to the exit transition output) stays closed for 300 ms after the vehicle has left the park slot.

2.4 Data recording

Data provided by the sensor are stored in a stand alone (i.e. non-integrated) datalogger, which is an autonomous device designed to record a large quantity of data and to transfer them on a standard PC throughout the serial port.

The datalogger used in our device is the *HoboEvent* produced by OnSet Computers. This datalogger is produced for environmental data collection, it has been used for example as a component in a rain gauge. It is not commonly used in automation devices, but it is appealing for its functionality, low cost, and robustness, especially with respect to environment agents. It is able to record up to 8000 events, comprising date and time in the following format: dd/mm/yy, hh/mm/ss. To record an event, it is necessary to shorten a contact between two metal wires coming out the rigid box containing the device. The Hoboevent is a completely autonomous device working with a proper CR 2032-lithium battery, for a time of 1 year of activity. It is possible to set some parameters of the Hoboevent through a special software package provided by the producer. One of the most important parameters is the option *ignore event*, through which it is possible to have the Hoboevent ignore events for a given amount of time after the contact shortening. This parameter has been used in our project, as explained below.

2.5 Circuit design

All the described components have been assembled as in the circuit represented in Fig. 3.

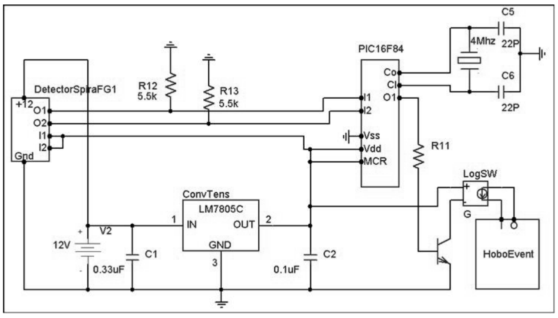

Fig. 3. The circuit scheme

Both the FG1 detector and the micro-controller circuit PIC 16F84 are powered by a 12V battery, through a proper voltage conversion. The voltage conversion is performed through an integrated transformer LM7805, which realizes a linear conversion voltage from a maximum of 30V to 5V stabilized. The two electrolytic condensers connected with ground are suggested by the producer (Texas Instruments) to stabilize the output and to reduce the overshoot phenomenon.

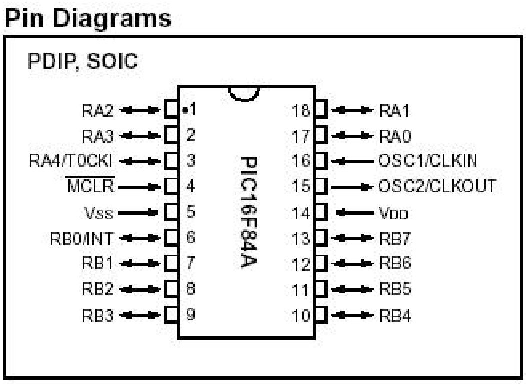

Fig. 4. Pin diagram

The two outputs of the FG1 detector are connected to the inputs of the PIC. The PIC requires a 4MHz quartz oscillator, considering that it is equipped with a prescaler, it actually works at 1MHz. The two condensers connected with mass complete the oscillating circuit. The PIC is obviously connected to the ground and to the 5V power supply. The MCRL pin works as a reset pin: when the voltage is high on this pin, the PIC starts to execute the firmware loaded in its flash memory. One of the output of the PIC should be connected to the relay contact of the Hoboevent. Actually, to prevent an excessive load to the PIC controller, this output pin is connected to the base pin of a BJT NPN which is used as an electronic switch.

Since the selected datalogger uses an electromechanical relay, the option *ignore event* of this unit has been set to 1-2 s to prevent the registration of fictitious events due the bouncing phenomenon existing in this type of relays. In fact, when the contact of the relay is closed, the relay shutter, which shorts the two output wires of the datalogger, could oscillate and generate a sequence of fictitious events. It is particularly important to take care of this undesirable phenomenon as the component gets older and older.

Two leds are placed on the control card. They give a fast representation of the controller state: if the green led is on, the park slot is free while, if the red led is on, the park slot is busy. Finally, if the green led is blinking the device is in the decisional waiting loop.

3. FIRMWARE DESIGN

The operation of the PIC16F84 control device follows the behavior of a two state Mealy automaton (see Fig. 5).

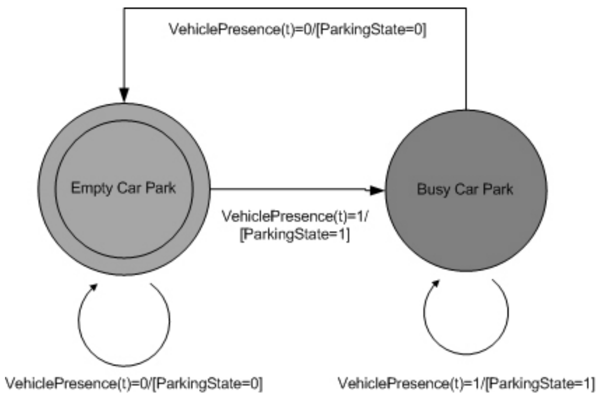

Fig. 5. Mealy automaton

The operation of the automaton represented in Fig. 5 is depicted in Table 1, where $S(t)$ represents the state of the automaton at time t and can assume two values: *Busy Car Park*, shortly *Busy* and *Empty Car Park*, shortly *Empty*. The input of the automaton at time t is denoted as *VehiclePresence(t)*, shortly *VPresence(t)*, while the output of the automaton is the *ParkingState*, shortly *PState*.

S(t)	VPresence(t) = 1	VPresence(t) = 0
Busy	Busy; PState=1	Empty, PState =0
Empty	Busy; PState=1	Empty, PState =0

Table 1.

The most important module of the firmware is the *state selector*, represented in the flow diagram of Fig. 6.

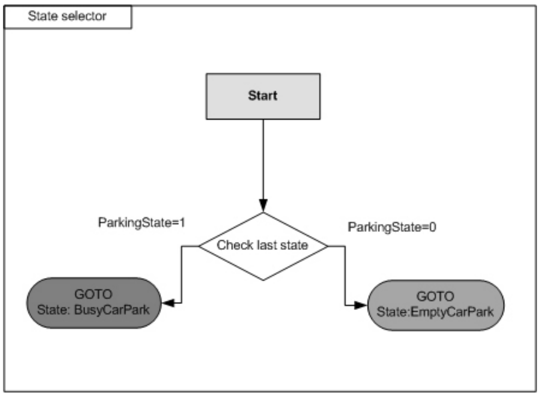

Fig. 6. State selector

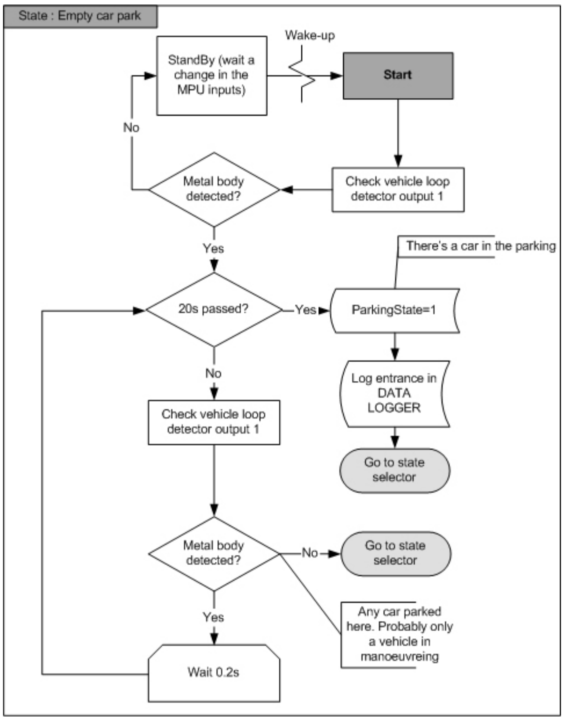

Fig. 7. Flow diagram for the state of empty parking

Such a kind of control paradigm, based on a Mealy automaton, is necessary since the datalogger is not able of detecting the type of the event but only records the closure of its relay.

The control algorithms implemented in the firmware are represented in the flow diagrams of Figures 7 and 8 for the Empty and for the Busy states, respectively. The idea of the procedure is as follows:

Empty State When a vehicle is entering the park slot:
- the Occupancy State is 0 (Empty State);
- the loop detects a metallic mass (i.e. Relays 1 is closed);
- the controller starts a 20s waiting period;

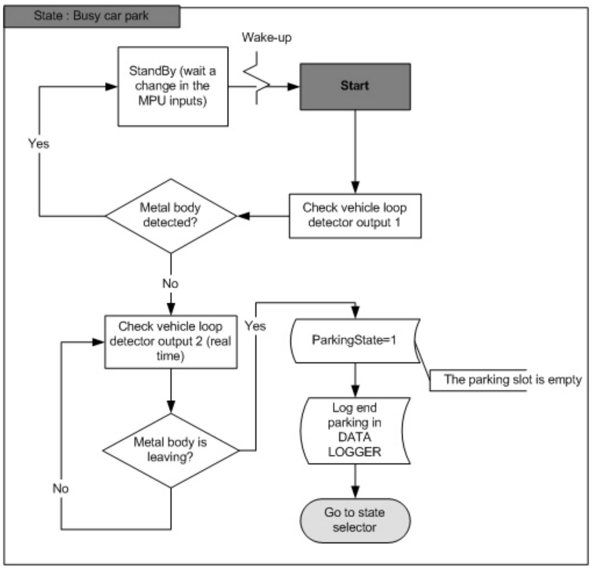

Fig. 8. Flow diagram for the state of busy parking

- if the vehicle remains for at last 20s,
  then an event is registered in the datalogger;
  the Occupancy State is set to 1 (Busy State);
  else no event is registered in the datalogger.

Busy State When the vehicle is leaving the park slot:
- the Occupancy State is 1 (Busy State);
- the loop detects the absence of metallic masses (i.e. Relays 1 is closed);
- the loop detects an outgoing metallic mass (i.e. Relays 2 is closed for 300ms);
- an event is registered in the datalogger;
- the Occupancy State is set to 0 (Empty State).

The part of the algorithm regarding the detection of loop in exit transition (which uses Relays 2 output) has been developed to avoid improper operations of the sensor.

Since the realized device should operate 24h per day for about one week, it is of interest to contain as much as possible power consumption. To do this, the control circuit remains active only for short time periods: 20s when a vehicle is entering the park slot and only for the time the vehicle requires to free the park slot. For the remaining time, the PIC is maintained in standby, in low power consumption mode, where the needed current is 1000 times smaller (from 2mA to 2μA).

3.1 The firmware programming language

The PIC is programmed through a RISC (Reduced Instruction Set Computer) assembler language specific of the Microchip.

It comprises 35 1-byte instructions (called *op*, *operation*) which are executed by the processor in one or two clock cycles. Since the processor works at 1 MHz, the execution of each instruction requires 1 or 2 μs. This is a negligible speed if compared to the clock frequency of modern processors, however it is completely sufficient to ensure the real time application we are interested in.

The PIC also comprises Ram and eeprom which makes it suitable in many application domains.

All processor family 16FXXX may be programmed by cheap serial port programmers and the assembler may be compiled using software available on the Microchip website. The firmware (compiled assembler) is then uploaded through a free software in the PIC flash memory.

4. RESULTS OF THE PROTOTYPE TESTING

The Sos.T.A. device (represented in Fig. 9) is characterized by a valuable precision in data detection. The large quantity of experiments showed that the device is able to recognize a parking event even if only half of the vehicle covers the parking slot area. If the sensor is located in the middle of the park slot, erroneous or improper vehicle positions can be over-passed. In a real urban scenario, where vehicles are positioned quite randomly with respect to the parking slots, the Sos.T.A. device correctly detected all the parking events, apart from a negligible number of errors mainly due to misplacements of the loop sensor or to battery exhaustion.

Fig. 9. The realized device

To reduce the detection error as much as possible, it is extremely important to select a proper value for the sensitivity of the loop sensor: a proper setting allows to exactly discriminate between park slot busy and empty and, in addition, to reject disturbances like motor-cycles standing or passing through the parking area.

Data can be analyzed through a processing data software developed in Borland CPP Builder on Win9x/NT/XP platforms.

In figure 10 it is possible to distinguish the interval of times where the parking slot has been busy, which are the black rectangles. At the base of each rectangle it is possible to read the starting and the ending times of each parking event. For convenience, a text box provides a list of all data collected. The realized software can manage more windows at the same time, allowing a fast comparison of data collected, and it is equipped with zoom and pan capabilities.

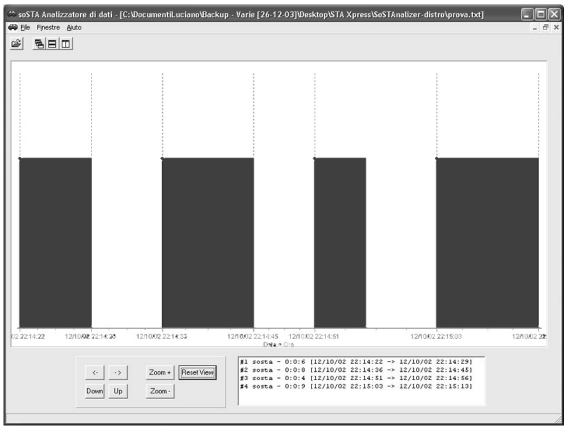

Fig. 10. The visualization window

5. CONCLUSIONS

In this paper a microcontroller based datalogger for parking data collection has been described. A loop vehicle detector has been selected as the most suitable sensor for this application. The described device satisfies all the specified requirements, namely the robustness, the low costs and the installation simplicity. As a future work, it is of interest to integrate the product and to develop a solar cell powered system connected to a data collection network.

REFERENCES

R. Luharuka and S. Krishnamurti. A microcontroller based datalogger for physiological sensing. In *IEEE instrumentation and measurement technology conferece*, Anchorage, Alaska, May 2002.

M. Matsuda. Traffic information collection systems powered by solar cells. In *1999 IEEE/IEEJ/JSAI International Conference on Intelligent Transportation Systems*, pages 870–873, 1999.