Design and Implementation of a Real-Time Tracking and Telemetry System for a Solar Car

Elie N. Mambou, Theo G. Swart, A.R. Ndjiounge, W. A. Clarke
Telecommunication Research Group
University of Johannesburg
P.O. Box 524, Auckland Park, 2006, South Africa
Email: {emambou, tgswart, arrichard, willemc}@uj.ac.za

***Abstract*—It is very critical for solar racers to achieve real time tracking of their solar vehicle during any race. The system is made of real time visualisation of the video feedback of the car from the escort vehicle, the location of the car on the maps as it is racing and its velocity. The real time video feedback of the race could be retrieved from the cloud for broadcasting purposes. The main advantage of this system is the wireless link communication which was achieved for distance transmission, coupled to a very reliable and robust desktop application for data sensor display purposes.**

***Index Terms*—raspberry pi, graphical user interface, global positioning system, inertial measurement unit, micro controller unit, controller area network**

I. Introduction

As fuel costs continue to increase, many questions arise concerning the future of combustible resources, that is why numerous automotive manufacturers are developing hybrid vehicles. At the same time many research are been conducted into alternative fuel vehicles [1].

The South Africa Solar Challenge (SASC) is a biannual event that happens in South Africa and is organised by the company called 'Sasol', which is one of the biggest fuel extractor firms in South Africa (SA). The aim of this challenge in to promote green energy, especially solar energy which nowadays, constitutes a reliable alternative source of energy. During the challenge, solar racers are called to compete along a well defined route in SA (see Fig. 1). The winner of the race is the team whose solar car has travelled the longest distance during the challenge [2].

During a solar challenge race, the solar vehicle is followed by an escort car. The role of the escort car is to observe and analyse the performance of the solar vehicle in order to monitor and optimise the race strategy. With the purpose of attracting a lot of visitors and raising awareness about solar energy during the SASC, the monitoring system should include real time video feedback of the solar-vehicle which can be accessible on the website, this is to allow solar energy followers to follow the performance of a solar car during the race. Furthermore a global positioning system (GPS) and inertial measurement unit (IMU) system is needed to accurately keep track of the motion of the solar car during the race. This research was done to improve the telemetry system of the University of Johannesburg (UJ) solar-powered car that was essentially based on global system for mobile (GSM) communication which was designed and provided by a base station network.

Fig. 1. SASC 2014 Route

The UJ solar team, after participating in the 2012 SASC with the famous Ilanga I solar powered car, opted to build a more advanced car called Ilanga II ("Ilanga" means "Sun" in one of the SA's official languages named isiZulu) for future races. This project involved a lot of students from different fields including industrial designers, electrical and electronic engineers and mechanical engineering students under the mentor-ship of some lecturers and companies.

Many research have been conducted by other solar teams around the world on the telemetry and tracking systems. The University of Arizona (UA) solar racing team has a telemetry system based on the XStream OEM Radio Frequency (RF) module, which is a RF module with reliable longrange wireless data communications [3]. This device allows an outdoor line of sight range of 7 miles, has a frequency of 900 MHz and receiver sensitivity of -110 dBm. The sender is on the solar vehicle while the receiver is based on the escort car. This allows the transmission of data via a wireless link. The main concern with this device is the throughput data rate of 9600 bps which may lead to data congestion during transmission. The Graphical User Interface (GUI) for the system was designed in Labview [3].

The Center for Product Design and Manufacturing (CPDM) of University of Malaya presented almost the same telemetry design as before. The sensors are connected to Input/Output (I/O) modules, the real time controller is responsible to process and send data to the Xstream OEM RF 900 MHz modules through a standard protocol for serial communication transmission of data, named RS232. The data was saved into compactRIO (real-time embedded industrial controller) as back up. The receiver was connected to the computer via a USB-RS232 cable for data acquisition [4].

The two previous systems are based on National Instruments (NI) solutions which are very expensive, not accessible by any team, only allow a small throughput data rate and do not achieve real-time tracking of a solar car. The system presented in this paper is based on cost-efficient electronics, with a total implementing cost of less than 150 USD, which is very low compared to other systems. Our system is very advisable for small teams or beginner solar racers in order to have a broad and detailed view on telemetry and tracking solar car systems. The main advantage of this system is the achievement of the real-time video feedback transmission via a wireless link from a Linux on-board computer to a Windows environment with a reliable throughput rate considering the huge data, which makes the tracking more robust, coupled to the live location of the solar vehicle on the map.

The rest of this paper is detailed as follow. In Section II, the tracking system design and implementation is presented. Section III shows the results and the last section deals with the issues encountered after the system deployment.

II. Tracking system design and implementation

The proposed tracking system is divided into 3 subsystems as shown in Fig. 2:

- Sensor board (SB): Its goal is to sense the position of the car during the race and send this data every 1 second to the chase car via wireless. The SB was designed in circuit design software called Altium Designer. It was important to design the SB from scratch because of low power consumption purposes as it has to dissipate as little power as possible from the batteries of the solar vehicle, which is not the case for most of the on-board computers on the market.
- Camera module (CM): The CM was mounted onto the Raspberry Pi (RPi) board which is programmed to send the video feedback of the solar vehicle to the chase car.
- Desktop application (DA): It is designed to provide a nice display of the video feedback as well as vehicle position on the map from a remote computer.

A. Sensor board design

The sensor board works as follows: The board is placed in the solar vehicle in a static position. The board is powered through controller area network (CAN) bus which is connected to the battery of the vehicle supplying a voltage of 12 V. The microcontroller unit (MCU), *PIC32M795F512L* from Microchip Technologies was used to synchronise data coming from the GPS chip and transmit them to the chase car via a wireless link. The purpose of some features on the sensor board and how they are interfaced with the MCU are detailed as follow.

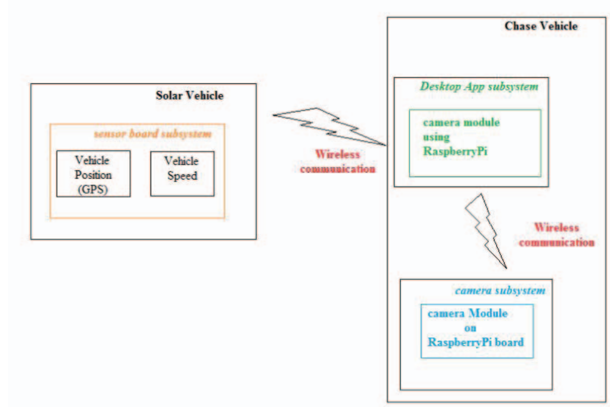


Fig. 2. Overview of the tracking system

- Voltage regulator circuit: It has a role of maintaining and regulating voltage level through the circuitry. In our case there are 2 voltage regulators. The board is powered from 12 V DC, the first regulator (LM2596SX-5.0) step-down the input voltage to 5 V (input voltage of USB port) and the second regulator (LM2596SX-3.3) step-down to 3.3 V (input voltage of GPS receiver, SD card socket, MCU).
- PIC32 connection: The basic connection for the PIC32 includes all VSS, AVSS pin to be grounded, similarly all VDD, AVDD pins should connected to 3.3 V. The ceramic capacitor on the board with a value of 100 pF are used to reduce power fluctuations throughout the circuitry. The PIC32 has an internal crystal of 8 MHz which is relatively low, a crystal oscillator of 28 MHz was added to provide more stable oscillation. The master clear reset pin (MCLR) is connected to a push button and a resistance of 500 Ω is placed in between to limit the current flowing through the line for safety purposes [5].
- In-dircuit debugger (ICD) connection: This is a port on the board which enables the programming of the chip, in our case an in-circuit debugger 3 (ICD 3) was used to debug the chip.
- GPS module: The NEO-6G GPS receiver was used. The most important connection on the GPS schematic is the transmission pin (TXD) of the GPS receiver which is connected to a reception pin (RXD) of the MCU. Because of the short range of the GPS receiver which could not give any reading indoors, a GPS antenna with a signal-to-noise ratio (S/N: 308400) was mounted on the board to increase the accuracy in data [6].
- Secure digital (SD) card connection: The SD card is connected to the MCU via an SD card socket. The purpose of the SD card was to save the positions of the vehicle in a text file given by the GPS sensor.
- Wireless module: A wireless transceiver called XBee S2 was embedded in the SB to achieve the transmission of data to the remote laptop using the protocol 802.15.4 at 2.4 GHz.

- USB Port: This port was an alternative way of powering the sensor board from 5 V in case that a 12 V power supply (battery voltage) was not available.

Fig. 3 presents the sensor board being programmed using MPlabX IDE. With the board connected to a 12 V DC power supply from CAN bus and programmed via an ICD3 debugger.

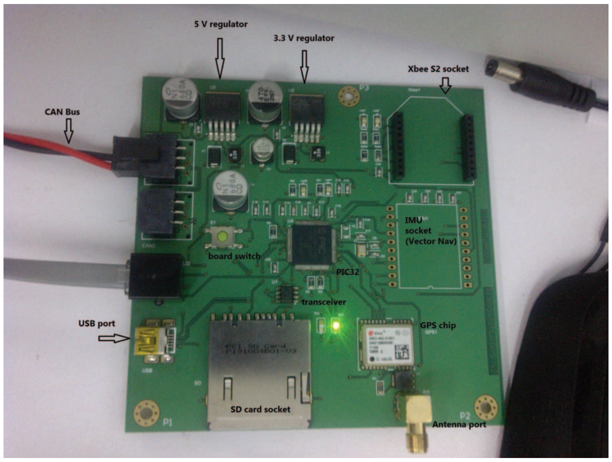


Fig. 3. Sensor board

B. Camera Module subsystem

Fig. 4 shows a snapshot of a digital clock video feedback taken by the RPi camera module mounted on the RPi board and transmitted via wireless to another computer through a unique internet protocol (IP) address and a communication port. As it can be seen on the snapshot, there was a delay of about 4 seconds in the transmission. This delay was due to the streaming of the video from receiver side through the port pipeline. The RPi was coupled with a Wifi dongle with the following properties: 2.4 GHz, 150 Mbps, 802.11 b/g/n Wifi.

C. Desktop application

This subsystem was based on the effective reception of these data, evaluating the delay in streaming and saving data. The desktop application was called "UJ Solar Car Video Tracker". This application was implemented using C♯ programming with Windows Presentation Foundation (WPF) in the Microsoft Visual Studio 2012 environment.

III. Results

A. Camera subsystem

The camera system was successfully implemented as follow: A camera module was mounted onto the RPi, the commands "sudo apt-get update" and "sudo apt-get upgrade" were executed in the Linux environment to update the firmware of RPi and install all necessary libraries. The library called Raspberry Pi Video (raspivid) was used to interact with the camera module [7].

The following code from a script file was parsed to perform a wireless transmission of the video feedback to the desktop application:

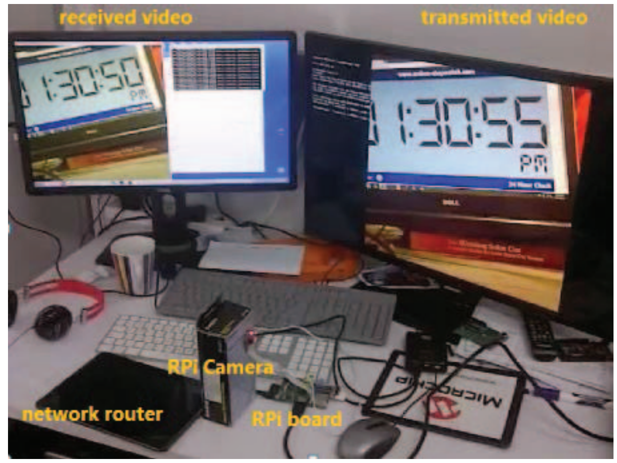


Fig. 4. Snapshot of RPi camera transmitting video

```
raspivid -width xx -height xx -t xx
-framerate xx
-output xx - || cvlc -vvv
stream:///dev/stdin -sout
# rtp {sdp=rtsp://portNumber }
```

Raspivid is the library for camera functions in RPi that includes the width, the height and the number of frames per second. On the other hand the output of the video is sent to a remote computer via a port number through a videoLan converter media player (VLC) file transfer protocol (FTP) pipeline. However on the remote computer, in order to access the video, the IP of the RPi is required as well as the same port number of communication.

B. Remote desktop application subsystem

Fig. 5 presents a screenshot of the desktop application running on a remote computer.

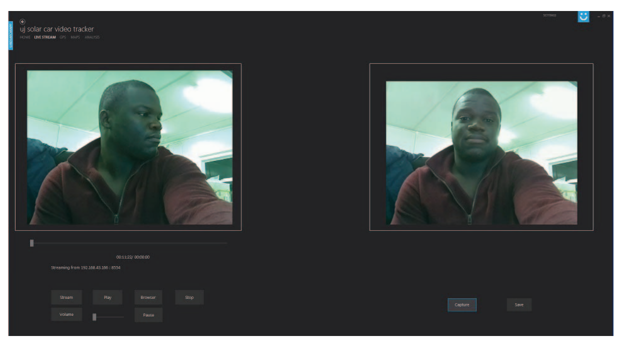


Fig. 5. Remote desktop application view of transmitted video

On the left-hand side, there is the video feedback retrieved from the RPi.

- Stream button: open the connection between the remote computer and the RPi board through the same port number.
- Play button: play of the real-time video.
- Pause button: pause the incoming video.

- Browser button: local browsing of any video for testing purposes.
- Stop button: stop the video.
- Volume gauge: adjust the volume of the streamed video.

On the right-hand side, the application allows snapshots saving from the live video. The button "capture" takes the snapshot of the current frame in the video viewer and the button "save" stores the snapshot taken in picture format in a browsed directory.

C. *Sensor board subsystem*

Concerning the sensor board, it was connected to a 12 V power supply and a ICD3 debugger in order to load the code into the MCU. The positions of the car by means of GPS coordinates were logged in a text file inside the SD card.

From the desktop application point of view, Fig. 6 presents the position of the car on the map from the GUI. This map was updated after every second to display the current position of the car and the coordinates were shown on the side on the application.

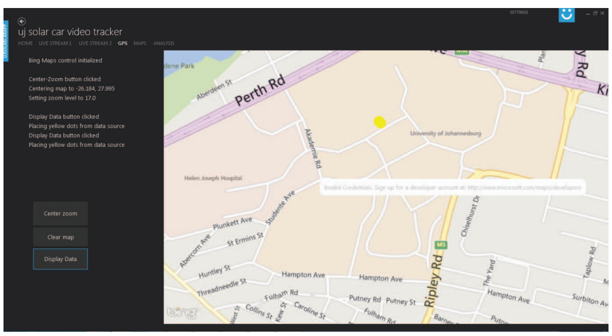


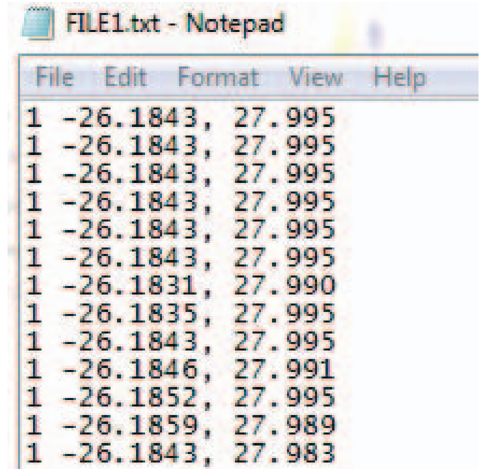
Fig. 6. Position of the car on the map

However the positions of the car were saved after every second from the SD memory of the sensor board located on the car. This was achieved before the data got transmitted to the escort remote application to be mapped.

The snapshot of the file containing some positions of the car is illustrated in Fig. 7.

IV. Deployment issues

The video feedback and the GPS data transmission is based on the Wifi protocol which is commonly used in homes. Wifi standards can divide Wifi signal in the 2.4 GHz block into up to 14 overlapping channels [8]. This kind of waves are short in terms of distance and have issues in penetrating solid objects or mediums like walls. The data reception within the system was disturbed around other networks or electronic devices which causes electromagnetic interference near the frequency of 2.4 GHz. There is a non-permanent signal when the sender and receiver are separated by huge obstacles like walls, buildings, etc. The signal is optimal when deployed outdoors and within the range of Wifi communication which is approximately 105 feet (indoors) and a bit greater outdoors.



FILE1.txt - Notepad

File Edit Format View Help

1 -26.1843, 27.995
1 -26.1843, 27.995
1 -26.1843, 27.995
1 -26.1843, 27.995
1 -26.1843, 27.995
1 -26.1843, 27.995
1 -26.1831, 27.990
1 -26.1835, 27.995
1 -26.1843, 27.995
1 -26.1846, 27.991
1 -26.1852, 27.995
1 -26.1859, 27.989
1 -26.1843, 27.983

Fig. 7. Logfile for GPS coordinates

A. *Interference*

In order to test the robustness of the Wifi transmission of the video feedback, a spectrum analyzer has been used to record interference signals at 5 different positions in the city of Johannesburg in the frequency range of $[2.462 - 2.482]$ GHz, which corresponds to the frequency band of our wireless adapter using the protocol IEEE 802.11b/g/n. The following results were recorded on the 11^{th} of August 2014 around noon time.

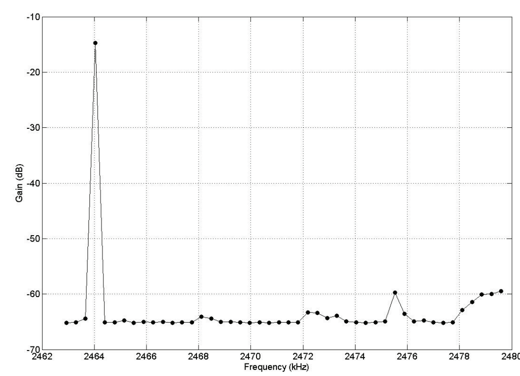


Fig. 8. Interference in Johannesburg at 26.024S, 28.012E

Figs. 8 and 9 present zones with low interference, the signal of transmission in those zones was very strong due to the weak magnitude of noise. While Figs. 10 and 11 show zones with a high level of Wifi signal turbulences, the communication was very disturbed through those zones because of the high noise which attenuates the strength of the communication, this leads to the loss of transmission signal.

B. *Doppler effect*

The escort car and the solar vehicle are in motion at different speeds while the wireless transmission of data is happening; therefore they are subject to the Doppler effect which is encountered when there is a change in frequency of a wave through a medium (or other periodic event) for

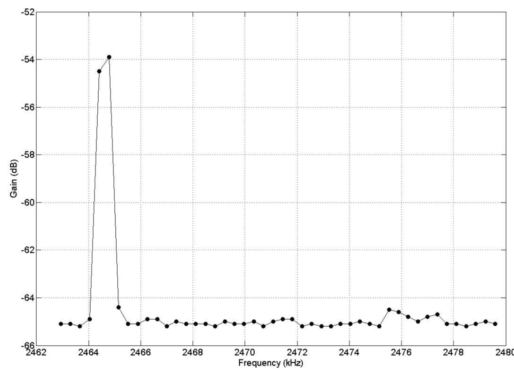


Fig. 9. Interference in Johannesburg at 26.108S, 28.054E

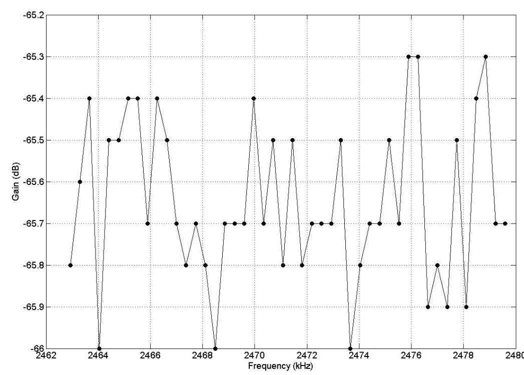


Fig. 10. Interference in Johannesburg at 26.259S, 27.903E

an observer (escort car) moving relative to a source (solar vehicle). The total Doppler effect is derived from motion of the source, motion of the observer, or motion of the medium. This causes the transmission to be attenuated because of the system subjected to relative speeds.

V. CONCLUSION AND FUTURE WORK

An approach to real-time tracking and telemetry system of a solar-powered car was proposed in this paper. The camera system was implemented successfully in the sense that the video feedback was sent over wireless through a static IP address and a port to a remote computer with a delay of around 3 seconds. This is a great achievement as a bridge was established between a Linux and a Windows platform to transmit a huge amount of data. The SB was designed, manufactured and assembled successfully. The GPS sensor placed on the SB was able to send GPS coordinates through the SB microchip. The data from that sensor was received after every second which is the frequency at which data was received from the GPS chip. The data was retrieved from a remote computer as well as the video feedback. Finally the desktop application was done, the video feedback from the camera module of the RPi was received as well as the positions of the car.

Nowadays, most of the camera surveillance systems are wired connected, which increases the cost of implementation

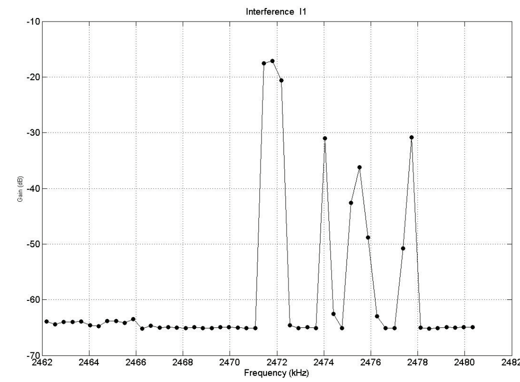


Fig. 11. Interference in Johannesburg at 26.166S, 28.038E

due to the distance of communication; therefore with our camera subsystem, a real-time camera surveillance can be installed in houses or companies or anywhere; all camera view within the system can be monitored from a remote server computer. The desktop application of the system which is very user friendly, after being installed on a Windows computer, can communicate with any Linux based electronic board via a unique port and a broadcast IP address given by the network.

Furthermore our proposed camera system can incorporate GPS data in the sense that every camera view is retrieved on the remote server computer with accurate GPS coordinates and time and therefore can be located on the geographic map. This provides a modern camera surveillance and security management system.

VI. ACKNOWLEDGMENT

The authors would like to thank the Solar team 2014 of the University of Johannesburg with their sponsors and mentors.

REFERENCES

[1] The syndicated, *A Technical Newsletter for ASIC and FPGA Designers*, Volume 5, Issue 1, Page 1.

[2] "Sasol Solar Challenge", Available: http://www.solarchallenge.org.za, [Online; accessed 14 September 2014].

[3] Timothy Baumgartner, Martin Bouvet Bours et al., *Solar Car Power Monitoring System Final Technical Report*, Submitted by Team 4431 on April 29, 2010. University of Arizona.

[4] Z Taha, R Passarella, et al., *Application of Data Acquisition and Telemetry System into a Solar Vehicle*, Computer Engineering and Applications Conference (ICCEA), 2010 IEEE International, 19-21 March 2010, pp.96-100.

[5] Microchip Technology Inc, *Microchip PIC32MX5XX/6XX/7XX*, 2009. Available: http://www.microchip.com, [Online; accessed 27 August 2014].

[6] u-blox, NEO-6, *u-blox 6 GPS Modules Data Sheet*, GPS.G6-HW-09005-E, Available: http://www.ublox.com. [Online; accessed 20-September-2014].

[7] Raspberry Pi, *Raspberry Pi Model B*, Available: http://www.rscomponents.com/raspberry, [Online; accessed 23-September-2014].

[8] Changwon Nam et. al., *Interference type based channel management using adaptive bandwidth in wireless LANs*, Personal Indoor and Mobile Radio Communications (PIMRC), 2010 IEEE International, 26-30 Sept. 2010, pp. 1360-1365.