

Computer Interface for Remote Teleoperated Mobile Robot Based on LabVIEW and Open Source Hardware

SCORC'CKO'G'J GTXG^{id}

National Higher Polytechnic School, University of Douala, Douala, Cameroon
University of Douala
Po. Box 1872, Douala
CAMEROON

Abstract: A remote controlled mobile robot utilizing National Instruments LabVIEW software, communicating over a low-rate serial ISM radio link. The robot is modular in design with system modules connected via serial interfaces to a system controller. A custom power supply and distribution circuit board provides power to the robot using sealed lead acid batteries as the source. The mobile robot streams system voltage and current status, JPEG images, ambient temperature and humidity readings, and GPS location. The project demonstrates the successful use of LabVIEW to develop a system using communication over lowrate serial radio links.

Key-Words: Global positioning satellites, power supply, radio transceivers, remote monitoring, robot control, seal lead acid battery

Received: June 25, 2021. Revised: March 17, 2022. Accepted: April 15, 2022. Published: May 7, 2022

1 Introduction

Remote controlled robots are not a new invention, there are many variations with differing designs. Many of these designs integrate their systems with complex text-based programming [1]. The robotic system in this paper is a LabVIEW based system. LabVIEW software uses a graphical programming approach making it easy to understand and learn. The remote-controlled mobile robot and remote base station were integrated with LabVIEW software. The mobile robot is modular in design and controlled by the system controller communicating to the robot modules via serial interface. Primary power source is 12 Vdc sealed lead acid (SLA) batteries, utilizing a low rate serial data link over industrial, scientific, and medical (ISM) radio band. A power distribution assembly provides 5 and 12 Vdc, along with current and voltage status. The mobile robot streams JPEG images via serial data link to a base station as visual aid for the user navigation and control. The mobile robot also transmits sensor data and results of data acquisition from internal sensors. The base station transmits motion command data and receives image and sensor data. A user interface displays the sensor data and sends commands to the robot. Cost of the project was a determining factor in selecting components.

2 Purpose

A robotic system is a very complex piece of machinery with numerous solutions for each component on the robot. We identified five engineering challenges to make a better robotic system. The following sec-

tions describe these challenges.

2.1 Modular Design

Previous robotic systems relied on specialized equipment. These robots relied on parts that are not easily replaceable without re-integrating their robot [1]. The robotic system in this paper, utilizes off the shelf equipment in a modular fashion.

2.2 Extended robot distance by using low data rate

The challenge of using the low data rate communication is the transmission of the visual data for the purpose of navigation and transmission of large amount of data obtained from the data acquisition. For applications where real time visual information is not the primary goal, low data rate interfaces can be used to report results of the data acquisition process. One example could be volcanic exploration [2], agricultural land monitoring where robot can be slowly preposition to different location by remote control or remote warehouse security monitoring [3].

2.3 Educational purposes

This system can be used for educational purposes as universal development platform easily scalable and adoptable to latest technologies. By utilizing only serial interfaces, students can develop autonomous robot sensors using Arduino modules [4].

2.4 Minimal firmware development

An objective of this project is to shift focus from the proprietary hardware design to modular, off the shelf

design solution scalable to many different software platforms easily exchangeable by using existing technology [5, 6, 8].

2.5 Simplified power supply and distribution

Other robotic systems utilize multiple voltages (12 Vdc, 9.6 Vdc, 7Vdc, +/-5Vdc) to operate their robots [1]. This is an added complexity that is not necessary with the wide use sensors, motors, and cameras that operate on 5 and 12 Vdc.

3 Solution

The project consists of two parts, a remote robot and local control interface as shown in Figure (1). The remote and local interface will communicate over the radio data link consisting of four different radio transceivers.

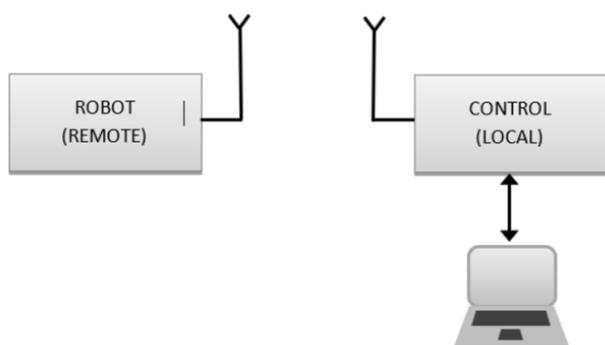


Figure 1: Simplified system block diagram.

The robotic system consists of the following sub systems: system controller, radio data link, power supply and distribution, motor control, and various sensors. Figure (2) is a block diagram of the simplified robotic system.

The local interface system is simple in design and consists of radio link section and GPS sensor. A personal computer is used as a system controller that hosts the user interface software. The simplified local interface block diagram in Figure (3).

3.1 System Controllers

All subsystems connect to a system controller using serial USB to RS-232 communication link. The advantage of this method is the modularity, module independence, and low cost. By using this method, modules, and system controller can be software and operating system (OS) independent. For the purpose of this project, we choose Windows 10 operating system with LabVIEW programming language. Each module that does not have a built in RS-232 interface

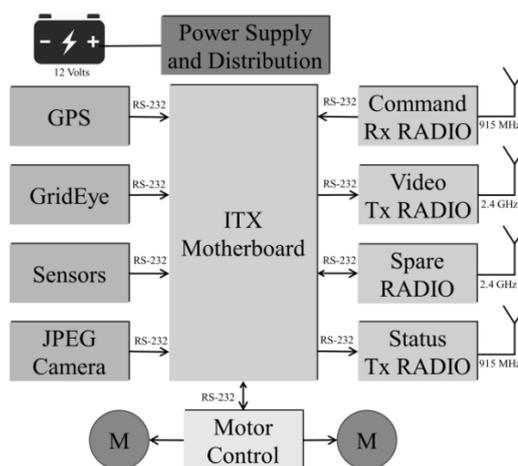


Figure 2: Remote interface block diagram.

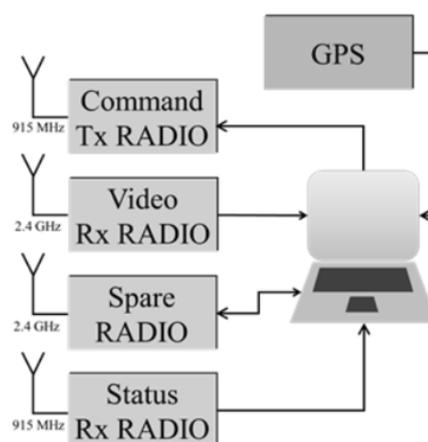


Figure 3: Local interface block diagram.

will use an Arduino Nano microcontroller board as the interface to the module hardware.

3.2 Data Link

Command data used to control robot motion will be transmitted over the command radio link. The command radio link will operate in the ISM band at frequency of 915 MHz and low data rate 19.2 kbps. In addition to command data, the link will request limited status data such as debug and error information from the robot.

Internal and external sensor data will be transmitted over the second ISM band radio link operating at 915 MHz and low data rate 19.2 kbps. Data from the JPEG camera and GridEye thermal sensor will be transmitted over a radio operating in the ISM band at frequency of 2.4 GHz at high data rate 115.2 kbps.

3.3 Power Supply and Distribution

A power distribution assembly PCB will provide power for the radios, sensors, controller, and motors. The power to the power supply and distribution PCB will be from an SLA battery. The power supply will convert 12 Vdc to 5 Vdc for the appropriate devices. The power distribution assembly includes current sensors, analog to digital (ADC) scaling circuitry and incoming voltage status provided to an Arduino Nano. Figure (4) is a block diagram of the power distribution assembly.

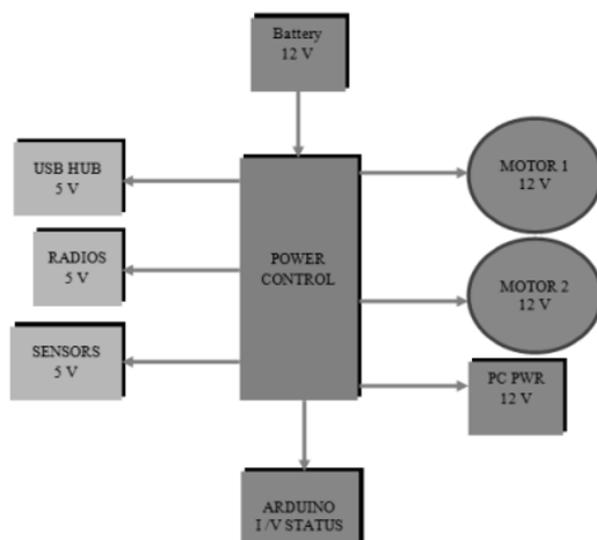


Figure 4: Power distribution block diagram.

3.4 Motor Control

The motor control module consists of two modules using L298N as a control driver and Arduino Nano microcontroller as a primary serial interface and message decoder.

3.5 Sensors

The following sensors will be utilized in the design to provide system status, robot awareness, and environmental status.

- (1) JPEG UART Camera
- (2) Panasonic GridEye Sensor
- (3) NEO M8N GPS Module
- (4) DH-22 Temp and Humidity
- (5) HC-SR04 Ultrasonic Sensors

4 EXTERNAL CONSTRAINTS

The LabVIEW based robotic systems design contains three external constraint: Federal Communications Commission (FCC) regulations, export controls, and environmental hazards. The following paragraphs describe these external constraints.

FCC regulations were identified as an external constraint for the radio transmission of the robotic system. These rules are in place for frequency spectrum management and human safety. By using predefined ISM frequencies there was no need for an FCC license.

Export control issues were identified for the JPEG camera and GridEye system. These regulations are in place to protect against the technology leaving the United States. By ensuring control over this technology, the robot will not be in export control violation.

Environmental hazards were identified for the SLA battery. The SLA battery has an environmental concern because it contains lead and sulfuric acid. It is important at the end of the batteries life that it is disposed of properly.

5 METHODOLOGY

5.1 System Controller

The main system controller on the robot side consists of the ITX Motherboard with the Solid-State Drive (SSD) running Windows operating system. The ITX motherboard is powered by a micro DC-DC converter supplied by the dedicated 12Vdc output from the power supply and distribution (PSD) PCB. The minimum operating voltage for the DC-DC converter is 10.8Vdc, which is below the minimum usable voltage for the SLA battery. The main system controller communicates with the robot modules using USB serial communication bus. Each module can be set to specific data rate, which can be configured using system configuration software written in LabVIEW as shown in Figure (5).

5.2 Data Link

The transceiver modules used as data links in this project are E44-TTL-1W and E39-TTL-100. The technical specifications of the radios are listed in Table I.

The transceivers operate on 5 Vdc and have a built in UART interfaces. The transceivers are connected to a protoboard containing a TTL to RS-232 converter with power supply terminals as shown in Figure (6).

5.3 Power Supply and Distribution

A design challenge of the robotic system was to simplify the power supply and distribution (PSD), while providing current and voltage status. To meet

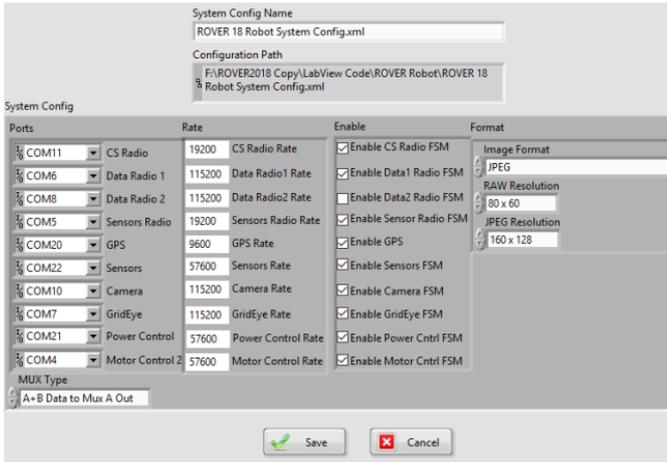


Figure 5: System configuration interface.

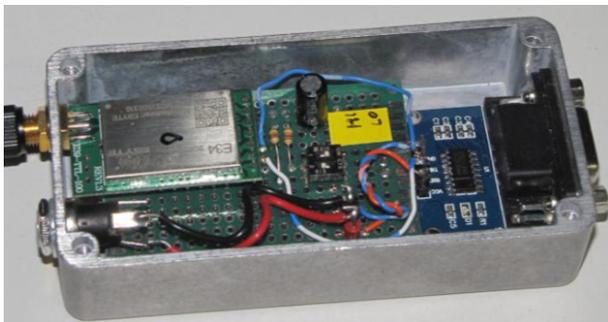


Figure 6: Radio module.

the challenge of a simplified PSD, we created a custom printed circuit board (PCB). The following paragraphs describe the design process of the PSD circuit board.

The first design decision was to determine the voltages to supply and distribute with the PSD PCB. Because the robotic system uses a 12 Vdc SLA battery as the source, it was determined to distribute 12 Vdc. The robotic system is modular in design and uses over eight USB interfaces. Because USB interfaces operate on 5 Vdc, we chose to supply 5 Vdc from the PSD PCB.

The next decision of the PSD PCB was how to convert 12 Vdc to 5 Vdc. The approach taken for the power supply was to use a hybrid power supply method. A hybrid power supply uses a DC-DC switching regulator in conjunction with a linear regulator. Using the hybrid power supply method, we were able to provide a stable 5 Vdc that is efficient with reduced noise. The DC-DC regulator selected for our power supply is a muRata OKR-T/10. The OKR-T/10 met all of our needs small in size, low temperature, and provides 6 Vdc at 10 A. The linear

regulator for the power supply is a MicroChip low dropout (LDO) MIC29300 providing 5 Vdc at 3 A. We chose to use three LDOs on the PSD to reduce heat, single points of failure, and to spread the load. Figure (7). is the schematic of the hybrid power supply with only one of the LDOs.

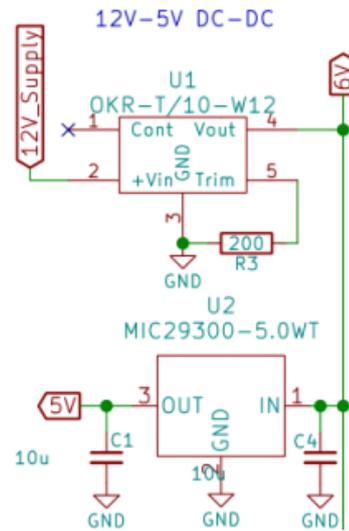


Figure 7: Hybrid power supply schematic.

To make the PSD unique we added features to monitor the voltage and current status of the robotic system. The method we used to monitor the robotic system currents is done with Allegro ACS723 Hall Effect current sensors. The ACS723 is an integrated circuit with a Hall Effect current sensor that provides a voltage corresponding to the current across the Hall Effect sensor. We chose to use the ACS723 IC because of its ease of use, fewer parts than a high side current sense configuration, and it is galvanically isolated. Figure (8) is a general diagram of the ACS723 depicting the Hall Effect with the voltage out.

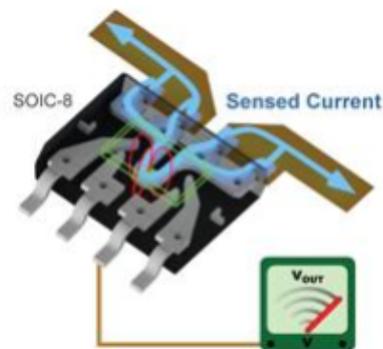


Figure 8: Hall effect sensor.

The ACS723-10AU has the ability to measure currents up to 10 A, this is over the 5 A we need for the 12 Vdc circuits and 3 A we need for the 5 Vdc circuits. Because of this we chose to scale the output of the ACS723 with an op-amp circuit as seen in Figure (9)., allowing us to scale the output of the ACS723-10AU from .5 Vdc to 4.5 Vdc.

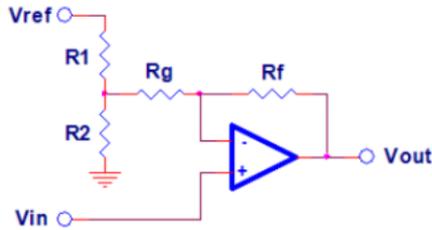


Figure 9: Scaling circuit.

$$V_{out} = m \times V_{in} - b$$

$$m = \frac{R_f + R_g + R_1 R_2}{R_g + R_1 R_2} \quad (1)$$

$$b = V_{ref} \left(\frac{R_2}{R_2 + R_1} \right) (R_f R_g + R_1 R_2)$$

The op-amp outputs are connected to an Arduino Nano. We chose to use the Arduino Nano as our ADC because there are eight channels, it is small, modular, and its USB output is easily integrated in the LabVIEW based robotic system. Figure (10). is the Arduino Nano used in our design.

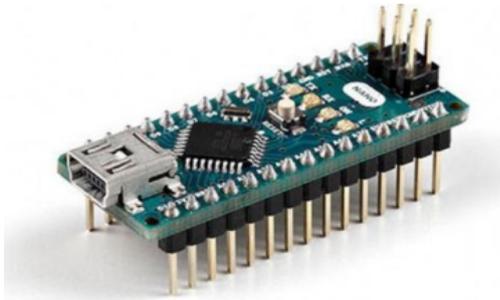


Figure 10: Arduino Nano.

The custom circuit board was constructed with special attention to board size, trace widths, heat sink placement, and isolation of the 12 Vdc distribution to the 5 Vdc supply. The first decision of the PCB was to determine the size. Based on the components size we determined the PCB needed to be 5" x 3". The next challenge was to determine the trace widths for the varying currents on the PSD PCB. Using the built-in trace width calculator in KiCAD we specified the trace widths according to the expected max currents.

The final challenge was to ensure the LDOs were on the outer edge of the PCB to accommodate for heat sinks. Figure (10) is the finished PSD PCB and Figure (11). is the fully populated PSB PCB.

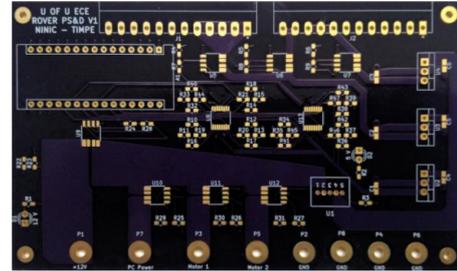


Figure 11: PS & D PCB.

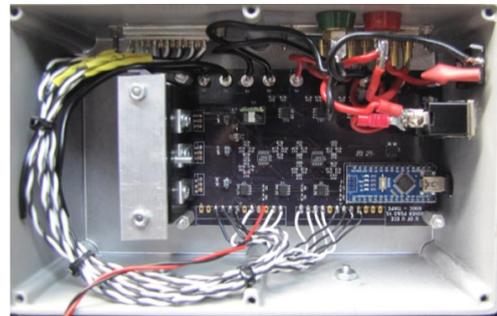


Figure 12: Assembled PS & D PCB.

5.4 Mechanical and Motors

The robot frame is built from the aluminum and steel railing in order to support weight of the battery and electronics components. The frame dimensions are 480mm length and 310 mm width. Figure (13). is the mechanical platform side view and Figure (14). the mechanical platform bottom view. Robot drive consists of two 12V dc motors attached on two 6-inch wheels and two 2-inch free spinning flywheels. Primary direction control is achieved by regulating left and right motor speed in direction of motion. With this approach, it is possible to achieve motion in any direction including 360-degree rotation around a central point.

To reduce cost, motors used in this project are recycled 12V windshield wiper motors modified to enable 360 Degrees of rotation.

5.5 Motor Control

The motor control unit section operates in the asynchronous mode in the following manner: When operator moves joystick, the input from joystick is decoded to calculate direction and speed based on the

Table 1: Measured Speed.

Wheel Radius(m)	s/rev	Direction	Speed (m/s)
0.089	1.624	(R)CW	0.3443
0.089	1.596	(L)CCW	0.3503
0.089	1.616	(L)CW	0.3460
0.089	1.644	(R)CCW	0.3401



Figure 13: Mechanical platform side view.

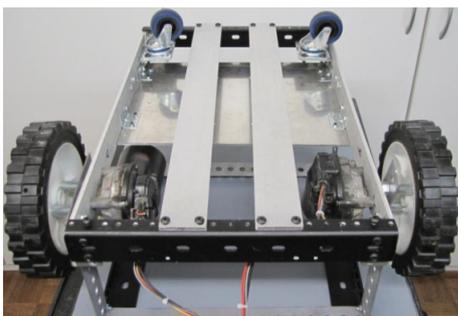


Figure 14: Mechanical platform bottom view.

joystick arm (x, y) location. Calculated direction and speed are then converted to string command and passed out to the CMD/STS radio to be transmitted to the remote location Figure (15).

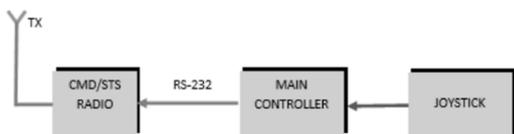


Figure 15: Local system motor command path.

This operation is repeated as long as operator controls the joystick. When the robot side receives the command from the local side, the software sends command to the motor controller unit (MCU) via serial

bus Figure (16). When motor MCU receives the command, it generates PWM voltage and drives the motor in the duration of 50ms and turns motors off without a valid command. The delay in the motor controller loop ensures motor continues to operate in the duration larger than transmission delay since delay exists between each consecutive moving commands. This delay can be adjusted to ensure continuous response and smooth drive.

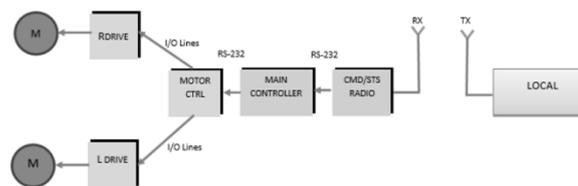


Figure 16: Remote system motor command path.

The MCU consists of Arduino Nano microcontroller board and two dual L298N H-bridge motor driver boards connected in parallel to increase driving current. MCU block diagram is shown in Figure (17).

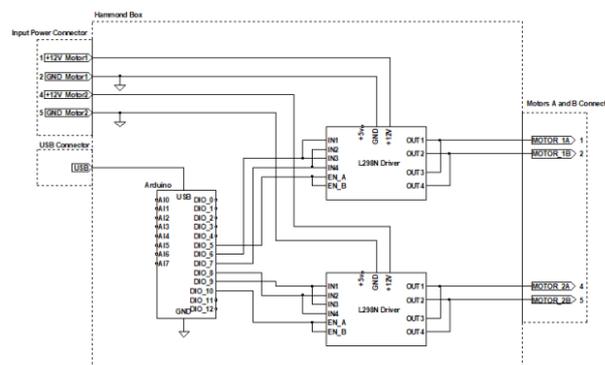


Figure 17: MCU block diagram.

Each motor driver board is supplied with a separate 12V dc line from the PSD PCB in order to monitor current draw for each motor. The assembled MCU is shown in the Figure (18).

The software for the MCU Arduino controller performs two functions. First function is to receive serial data command using Arduino serial RS-232 port operating at 57.6kbps and to decode command syntax. Second function is to generate PWM pulse for each motor driver using Arduino “analogWrite()” software function. The frequency of the PWM signal is approximately 490Hz and the duty cycle will depend on the motor speed parameters in range from 0 to 255. Figure (19). Is the basic software algorithm for the

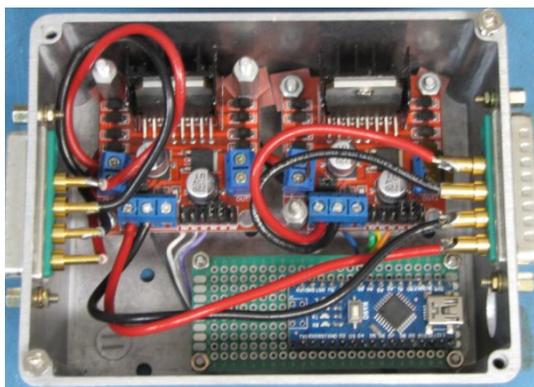


Figure 18: Assembled MCU

of JPEG images at 115.2kbps data rate that can be sent over the radio link with the same data rate, as shown in Figure (20).



Figure 20: uCAM-III.

motor control

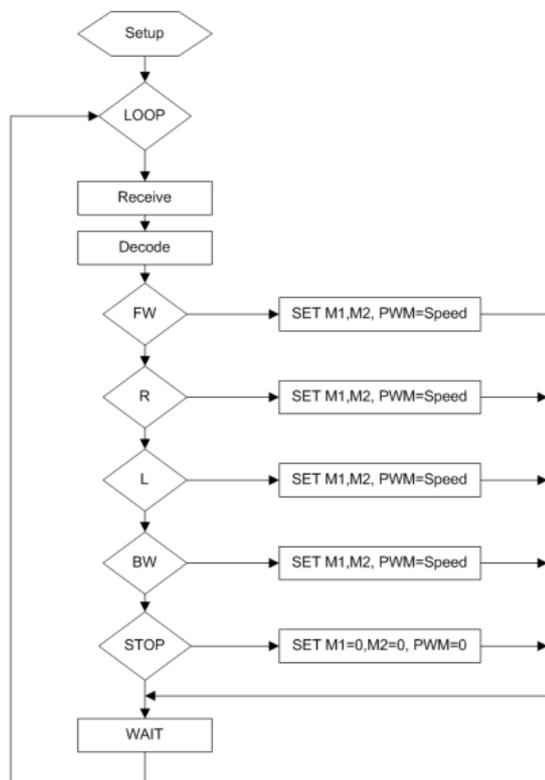


Figure 19: Motor control firmware algorithm

The software will also enable or disable control pins on the L298N H-bridge motor driver boards based on the command syntax direction parameters.

5.6 Sensors

5.6.1 Serial Camera

The camera module used in this project is the uCAM-III serial camera module with built in UART interface. This camera was chosen for its ability to stream series

The camera is supplied with the 5Vdc from the PSD PCB and it is connected to the USB serial bus via RS-232 TTL to serial converter board. Camera is set to operate in the JPEG picture mode with 80 x 60 resolution. In order to interface camera with the main controller software, LabVIEW driver is developed to utilize all functions necessary for camera operation.

5.6.2 Grid-Eye Thermal Image sensor

The second visual sensor used on the robot side is the Panasonic Grid-EYE 8x8 Infrared Array Sensor.

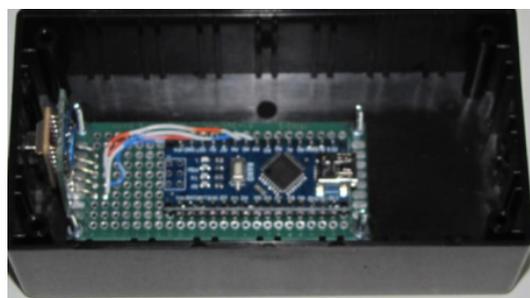


Figure 21: Grid-Eye sensor (left) with the Arduino controller.

This sensor was chosen as a simple low cost thermal imager which can provide option for temperature detection with 64 pixel of resolution. This sensor is connected to the Arduino via its I2C interface and it is supplied with 5Vdc from the Arduino board. The software on the Arduino board interfaces with the Grid-EYE sensor and it converts thermal image pixel array data into serial data stream at 57.6kbps data-rate. Serial data is then passed asynchronously to the main controller software via USB serial interface. The start of the data is prefixed with the "Grid-EYE:" word following by the 64 elements of thermal

data and it is terminated with the “End” word. The Grid-EYE Arduino controller streams serial data as soon as it is powered up. For this sensor a LabVIEW driver was developed to interface with the serial data stream in the main controller code

5.6.3 GPS Sensor

The robotic system utilizes two GPS sensors one on the robot and another on the local station. The GPS sensor allows for latitude and longitude status of the robot and local station. Figure (22) is the NEO M8N GPS sensor integrated on the robot.

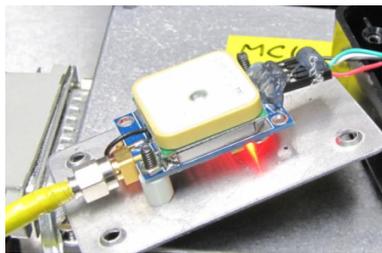


Figure 22: GPS sensor.

5.7 Robot Software

The robot controller software is written in LabVIEW programming language deployed as executable package running Windows operating system. The robot software consists of eight finite state machines (FSM) running in parallel and producer consumer loops executing commands from the command status radios. Each FSM controls one hardware module via serial bus. Data between FSM is shared using global variables and data queues.

The software configuration such as data rate, COM ports and module enable function can be changed using test system settings user interface shown in Figure (5). The configuration data is stored in the xml file and it is read by the robot software upon startup. Since software operates on the robot side, there is no user interface associated with the software.

5.8 Control Software

GUI The graphical user interface (GUI) runs on the control side of the project. It consists of 5 FSMs running in parallel, User event loop, image data processing loop, sensor data loop, PRBS counter and debug loop. Data is shared between the loops by using global variables and queues. The control GUI has dedicated system configuration similar to the robot side. The front panel of the control GUI is shown in Figure (23). The front panel interface consists of following functional blocks:

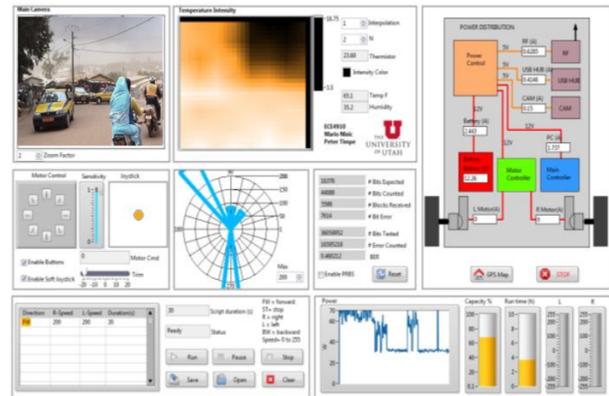


Figure 23: Control GUI.

The JPEG image block (top left) displays image received from the robot side and thermal image window next to it displays the thermal image received from the Grid-EYE sensor. The JPEG camera and thermal image sensor are positioned next to each other on the robot side and the received thermal image represents low-resolution thermal image of the received JPEG picture.

The power distribution block (top right) represents all measurements collected by the power board. Each measurement is placed next to the simplified block diagram representing robotic vehicle power distribution. The motion control block with the Scripting engine enables user to command robot motion in desired direction. User have the option to move robot by use of hardware joystick or by predefined direction controls. The scripting engine (bottom left in the Figure (23)) enables user to write simple scripting commands and to move robot by executing those commands. The syntax for the motion commands consists of the five basic direction command words: forward FW, backward BW, Right R, left L and stop ST. There are two speed parameters ranging from 0 to 255 for each motor and one duration parameter in seconds. The speed parameters are used in similar fashion as described in the Motor control section and duration parameter specified the duration of execution time for the current script line. The command for the forward motion with maximum speed in the duration of 5 seconds will look like example in Figure (24). There is also an option to load and save script in the text format.

The distance sensor chart (middle of the GUI) displays the distance measurements in centimeters for each ultrasonic sensor described in the section above. The BER test block (g) enables user to switch robot data transmission from JPEG image to Pseudo-Random Binary Sequence (PRBS) pattern for use in data transmission debug. When enabled, this function will send instruction to the robot to start transmit

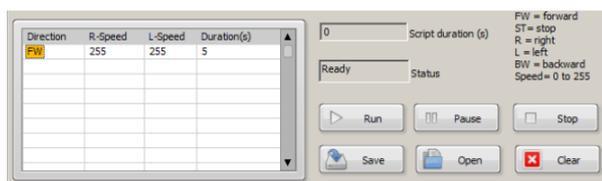


Figure 24: Script example.

ting 2047 samples of PRBS data instead of JPEG data. User can use this data to estimate quality of the data link based on the Bit error rate calculated based on received PRBS data and internally generated PRBS data. The power indicator graph located on the lower right corner of the GUI in Figure (23). is the visual indicator of the power consumed by the robotic vehicle. Another feature of the GUI is the ability to display both GPS locations of the robot and base station if the local control computer has available internet connection.

6 Results

The robotic system we designed is fully functional and operates within the guidelines we set at the start of the project. Figure (25) is the final robotic system and Figure (??) is the location station. The following sections list all the functional equipment associated with the robotic system.



Figure 25: Completed robotic system.

6.1 Software

- (1) Successfully implemented LabVIEW software on the local and robot system controllers.
- (2) Fully functional user-friendly command and status GUI.

- (3) GUI provides joystick control and preplanned script control.
- (4) GUI offers the ability to map the robotic system and local GPS locations (see Fig. 25.)

6.2 Sensors

The following sensors are all fully functional on the robotic system.

- (1) JPEG UART Camera
- (2) Panasonic GridEye Sensor
- (3) NEO M8N GPS Module
- (4) DH-22 Temperature and Humidity
- (5) HC-SR04 Ultrasonic Distance Sensors

6.3 PS & D

- (1) Provides 5 Vdc to the sensors and radios.
- (2) Provides 12 Vdc to the Motors and ITX motherboard.
- (3) Fully integrated Arduino Nano on the PCB.
- (4) Monitors current from 0 to 5 A on the 12 Vdc lines (SLA batteries, motors, and ITX motherboard).
- (5) Monitors current from 0 to 3 A on the 5 Vdc lines (radios, sensors, and USB hub).
- (6) Measures the SLA batteries voltage.
- (7) 12 Vdc lines can use up to 5 A.
- (8) 5 Vdc lines can handle up to 3 A.
- (9) LED status of 5 Vdc and 12 Vdc.

6.4 Mechanic and Motor Control

- (1) Robot can rotate 360 degrees in a 1 m radius.
- (2) Functional PWM motor drive.
- (3) Maximum motion speed .3 m/s.

6.5 Conclusion

The purpose of this project was to design and build a working prototype of the robotic vehicle developed using LabVIEW programming language and controlled by the serial communication radio link operating in ISM band and using minimal budget. We successfully developed a robotic platform that operates entirely by the LabVIEW programming language controlling the system, operation, and communications. We were able to integrate all peripheral modules using only serial communications. The robot

motion control and sensor communication link performed correctly at a distance of 450 m. Overcoming the challenges of transmitting image data JPEG data format over a low data rate, were the loss of bit results in the loss of the image we were able to achieve a distance of 300 m. We also developed a simplified custom power distribution board with voltage and current sensors for the system operation monitoring and control.

References:

- [1] J. Grimes, F. Kapsenberg, C. Hyman, T. Morton, T. Slone, B. Porter, J. Hortnagl, R. Felber “2011 Oregon State Mars Rover Design Report”, Oregon State University Robotics Club, Oregon State University, Corvallis, Oregon, USA.
- [2] D. Caltabiano ,G. Muscato (July 1st 2005). “A Robotic System for Volcano Exploration, Cutting Edge Robotics”Vedran Kordic, Aleksandar Lazinica and Munir Merdan, IntechOpen, DOI: 10.5772/4666.
- [3] P.Thirumurugan, S.Sivakumar, S.Arunrajan, S.Nagaarjun, K.Prakash, R.Shivamanikandan,“Labview Based Surveillance Rover in Military Applications”, International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, Vol. 7, Issue 2, February 2018.
- [4] A. A. Galadima, “Arduino as a learning tool”, 2014 11th International Conference on Electronics, Computer and Computation (ICECCO), Abuja, 2014, pp. 1-4.
- [5] Y. S. Lee, B. Jo and S. Han, “A Light-Weight Rapid Control Prototyping System Based on Open Source Hardware”, in IEEE Access, vol. 5, pp. 11118-11130, 2017.
- [6] A.H.Samba , A.Yeremou Tamtsia , L.Nneme Nneme, I.J.Hermine Som ,and A.Houwe, “Design and Implementation of Wireless Iterative Learning Fault Estimation Algorithm with quantized measurement and Packet Losses”; IEEE Access, vol. 18, no. 1, November, 2021.
- [7] S. A. Herve, Y. T. Aurelien, H. S. I. Judith, and N. N. Leandre, “Networked iterative learning fault diagnosis algorithm for systems with sensor random packet losses, time-varying delays, limited communication and actuator failure: Application to the hydroturbine governor system,” WSEAS Trans. Syst. Control, vol. 16, no. 20, pp. 244–252, May 2021, doi: 10.37394/23203.2021.16.20.
- [8] S.D. Gadzhanov, A. Nafalski ,” LabVIEW Based Remote Laboratory for Advanced Motion Control”. 2014 IEEE, 11th International Conference on Remote Engineering and Virtual Instrumentation, 2014.
- [9] Z. Nedic, "A Universal Workbench for Motion Control Experimentations in LabVIEW Environment," in 9th International Conference on Remote Engineering and Virtual Instrumentation (REV), Bilbao, Spain, 2012, pp. 51-57.

Creative Commons Attribution License 4.0 (Attribution 4.0 International , CC BY 4.0)

This article is published under the terms of the Creative Commons Attribution License 4.0

https://creativecommons.org/licenses/by/4.0/deed.en_US