University of Toronto
Edward S. Rogers Sr. Department of Electrical & Computer Engineering
ECE532H1 S: Digital Systems Design

University of Toronto Search and Rescue Robot
Group Final Report

Prepared by:
Ahmed Sageer Cheriya Melat
Bojian Zheng
Kejian Li
Jinghan Guan

Contents

1. Overview

This section provides the project overview of the design UofT_SAR_Bot.

1.1. Project Description

University of Toronto Search And Rescue Robot (*UofT_SAR_Bot*) is a Wheeled Mobile Robot (WMR) designed on FPGA platform. This four-wheels-driven rover is a prototype model of robots intended to assist human beings in rescue and safety operation in environments that are not easily accessible to human beings.

UofT_SAR_Bot has two modes of operation, namely automatic (‘auto’) and manual (‘man’) mode. These modes are selected from the android-based remote control device. In ‘auto’ mode the robot emulates the behavior of detecting a target and going towards it automatically. In the project, the target is a small rectangular blue screen. With the use of color filtering and blob detection, the robot can detect the target and move towards it until the robot is sufficiently close to the target. During this target-following operation, the robot makes use of information from range-finders to avoid dangerous collisions and to turn away from any static or dynamic obstacles that block the path to the target. With suitable sensors like passive infrared sensor (PIR) the robot could be modified to detect an object that emits thermal energy, potentially a breathing human being. In ‘man’ mode, the robot can be manually maneuvered in any direction with any speed selection from the remote-control device. This makes it possible for the users to send the robot to any area that is monitored through external surveillance cameras, but is hazardous for people to access it.

The basic automatic operation is pictorially shown in the Figure 1.1.1.

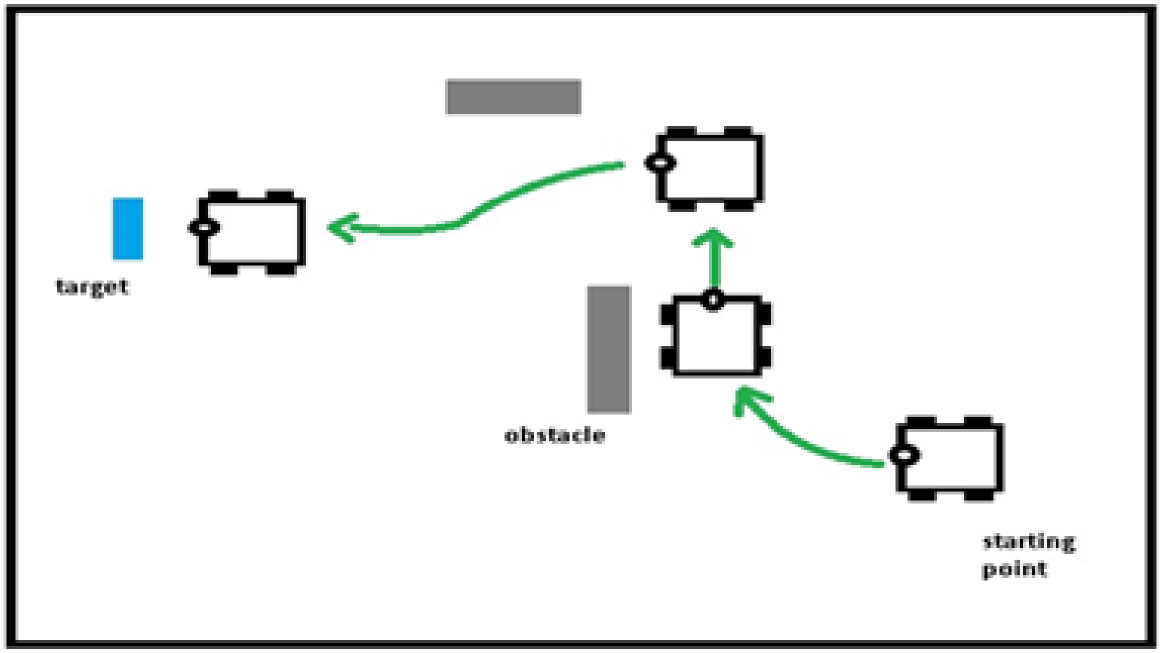


Figure 1.1.1. Robot moving to the target avoids an obstacle and takes an alternate possible path.

The remote-control user interface displays readings from sensors like temperature sensor, accelerometer, and gyroscope.

1.2. Motivation

The motivation of this project is to explore the possibility to utilize FPGA as central control of a WMR and provide hardware level acceleration to on-board computation tasks, especially in the image processing area. FPGA has shown great capabilities and potential in image processing. WMRs are otherwise normally implemented using microcontrollers.

Numerous practical applications that can be accomplished with WMRs are also very motivational. The proposed model of the robot can be a prototype for various practical application in search, rescue, and surveillance. One example of an application that looks very promising is the use of such WMR to go through small opening and passages through rubbles after a natural calamity in search of surviving humans, which otherwise is humanly uneasy. Another good application is the use of such WMRs to do surveillance and monitoring of hazardous area like a nuclear or chemical plant to collect information of the plant environment remotely without having much human exposure to radiation and noxious gasses.

Finally, the design team also sees this prototype as a good base model for various academic use. For example, with an addition of encoders and stereo cameras the robot model can be used for studies of advanced mobile robotics autonomous behaviors like localization, mapping, etc.

1.3. Project Goal

The goal of the project is to have a WMR designed using Digilent Nexys 4 DDR FPGA board. The four-wheels-driven WMR is expected to have a remote-control device through which users can select automatic or manual mode. In automatic mode, the robot should detect the target object and move towards it simultaneously avoiding the obstacle blocking the path. In manual mode, users should be able to remotely maneuver the robot and select the speed of the robot. At all times, the users should be able to remotely monitor the measurements from the robot, like acceleration, angular velocity, and temperature.

1.4. System-Level Block Descriptions

The physical model of *UofT_SAR_Bot* is shown in Figure 1.4.1. and the top-level architecture is shown 1.4.2.

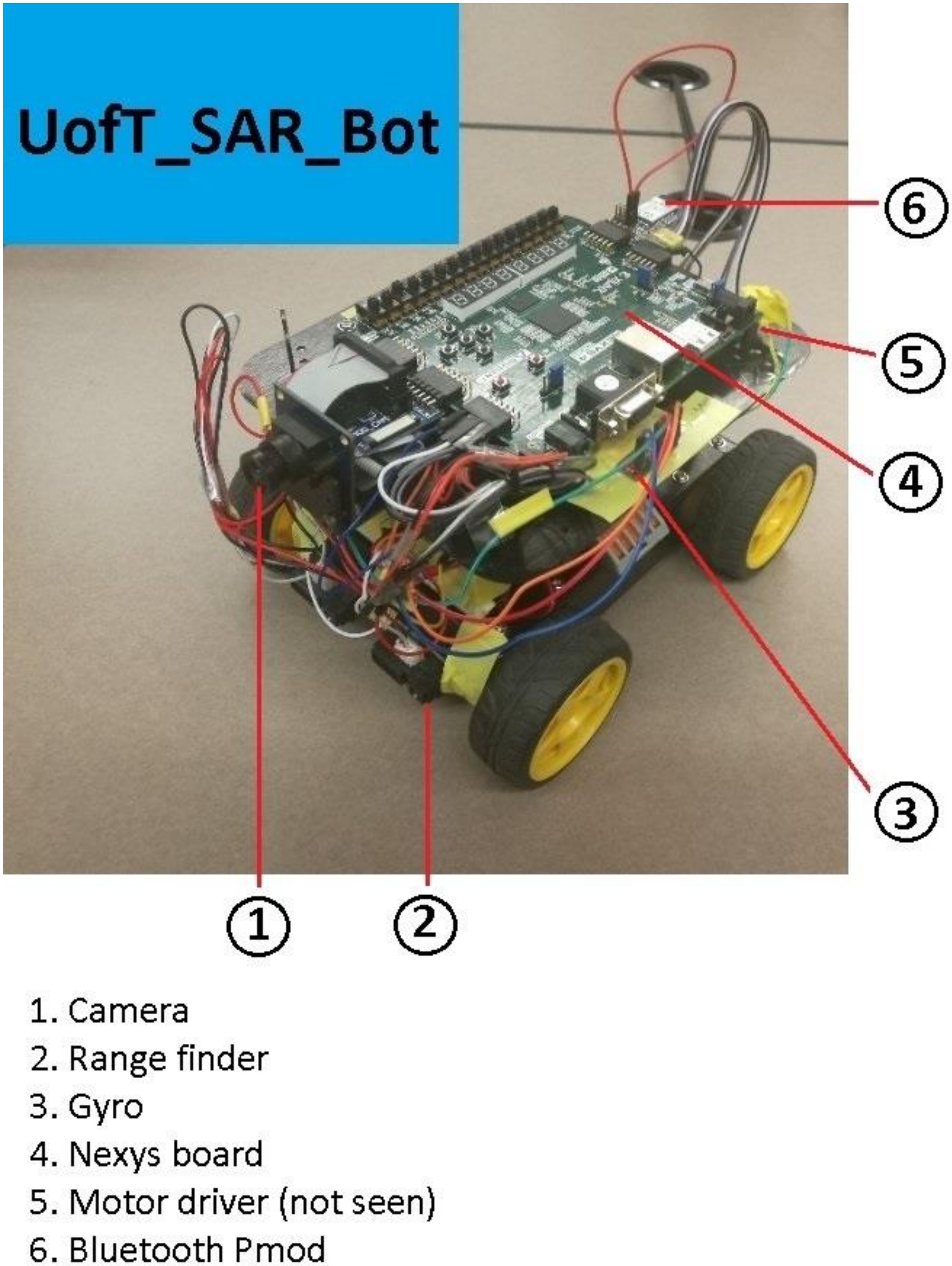


Figure 1.4.1. *UofT_SAR_Bot* Physical Model

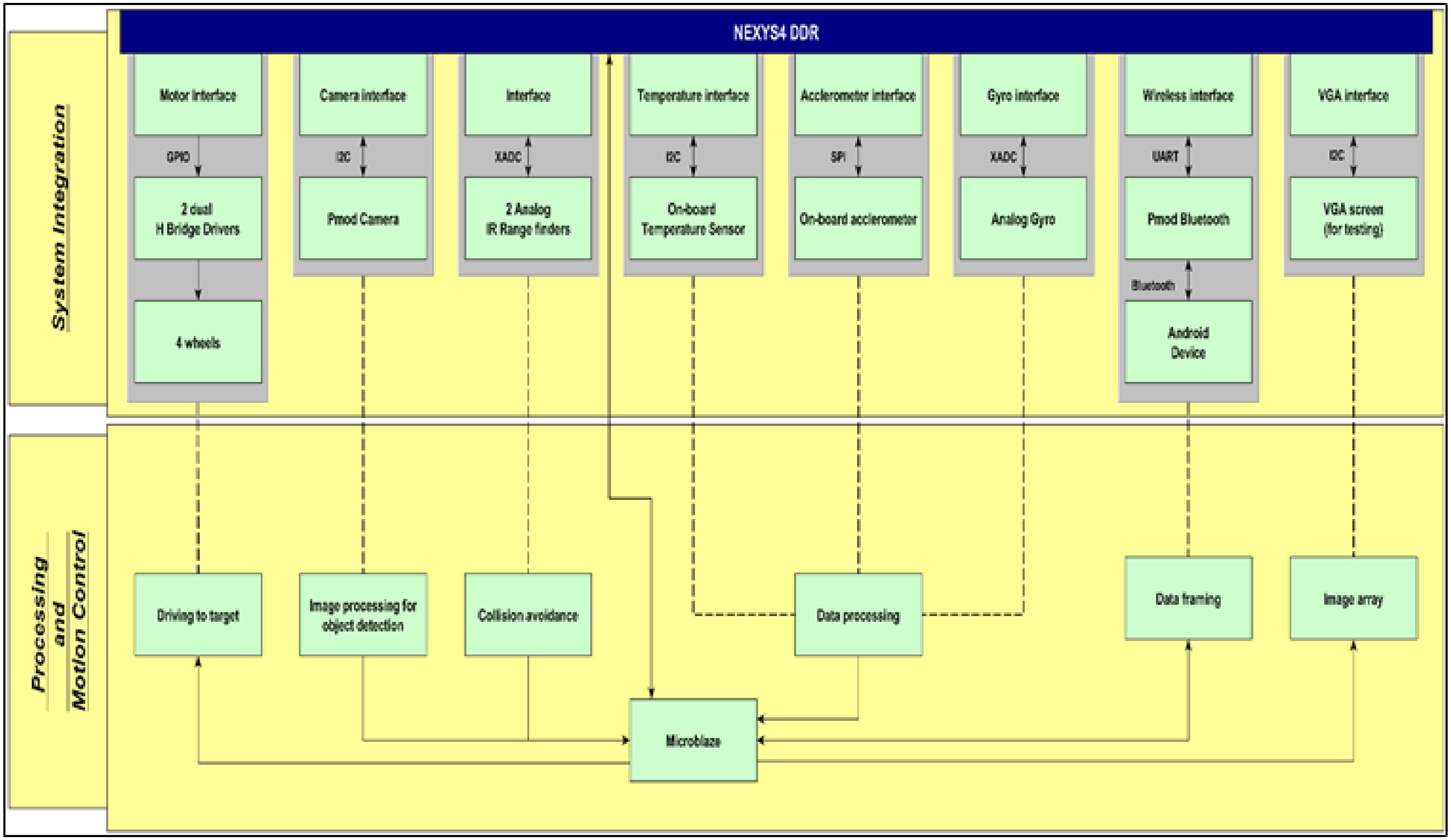


Figure 1.4.2. *UofT_SAR_Bot* Architecture using Nexys 4 DDR Board

The above architecture shows the overall design of the robot using Nexys 4 DDR board. The architecture is segmented into two layers. The upper layer 'System Integration' is showing the integration of each sensor/actuator or a user interface to the FPGA board. The other layer 'Processing and Motion Control' contains block that typically represents tasks performed (image and data processing and motion control) using the devices interfaced to the board. The block here may be an IP, peripheral and/or C programs on Microblaze. In the figure, solid lines represent hard wire connection or communication channel; dashed lines represent relation links.

1.4.1. System Integration

- FPGA Board (Figure 1.4.1.1.): Digilent Nexys 4 DDR board is the FPGA board used in the design. It comes with primary FPGA IC Artix-7 (XC7A100T-1CSG324C from Xilinx). Link to the device manual: https://reference.digilentinc.com/reference/programmable-logic/nexys-4-ddr/reference-manual.

Figure 1.4.1.1. Nexys 4 DDR Board (figure comes from [1])

- Wheels (Figure 1.4.1.2.): Of-the-shelf robot chassis from DF robot comes with four standard wheels with each running on DC motor with no-load speed of 200 rpm at 6 V. The wheels are interfaced to the FPGA board through two quantities of dual H-bridge drivers connected to port `JD` on Nexys board. Each H-bridge driver controls two wheels. The dual H-bridge motor driver used in the design is L9110 Dual Motor Driver operated at around 6 V DC. The driver module communicates to the FPGA board through GPIO. Link to the device manual: http://www.robotshop.com/media/files/pdf/instruction-man nual-rob0003.pdf (Robot chassis); https://www.creatroninc.com/upload/L9110%20Datasheet.pdf (Motor Driver).

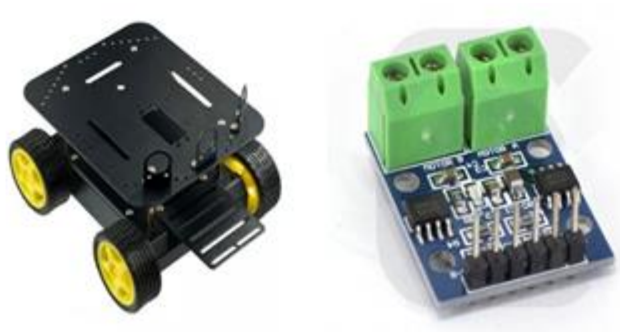


Figure 1.4.1.2. Left: Robot Chassis; Right: Motor Controller
(figure comes respectively from [2] and [3])

- Camera (Figure 1.4.1.3.): Pmod camera used is Omnivision OV7670 CMOS-sensor-based VGA camera. It is connected to port `JA` and `JB` on Nexys 4 DDR board. The image from the camera used in the design is RGB555 of size 320×240. The camera module communicates to the FPGA board using I^2C protocol. The captured image is put on 76800 deep 16-bits BRAM on the Nexys 4 DDR board. Link to the device manual: http://web.mit.edu/6.111/www/f2015/tools/OV7670_2006.pdf.

Figure 1.4.1.3. Camera Top and Bottom View (figure comes from [4])

- Range-finder (Figure 1.4.1.4.): The robot uses information from range-finder to know the distance to obstacles to avoid collisions. Range-finders used in the design give analog outputs. Two infrared range-finders are used in the design. The range-finder model used is Sharp GP2Y0A41SK0F. It has a range of 4 to 30 cm. The device operated at 5 V DC producing an output voltage of 0 to 5 V depending on the distance measured. Range-finders communicate to Nexys board through on-board XADC on port `JXADC`. To allow this, the output voltage of the range finder is scaled down to 0 to 1 V DC using external voltage divider circuit. Link to the device manual: https://www.creatroninc.com/upload/GP2Y0A41SK0F%20Datasheet.pdf.

Figure 1.4.1.4. Range-Finder (figure comes from [5])

- Temperature Sensor: Temperature sensor on the Nexys 4 DDR board is used to measure the temperature in which the robot is operating. The temperature sensor on board is ADT7420 from Analog Devices. It has a range of -40 to 50 °C. ADT7420 communicates to the FPGA using I^2C protocol. Link to the device manual: http://www.analog.com/media/en/technical-documentation/data-sheets/ADT7420.pdf.
- Accelerometer: Micro-Electron-Mechanical System (MEMS) accelerometer on the Nexys 4 DDR board is used to measure the acceleration of the robot along x-axis and y-axis. Since robot is operating on ground acceleration along z-axis is not measured. The accelerometer is ADXL362 from Analog devices. It communicates with FPGA through SPI communication protocol. Link to the device manual: http://www.analog.com/media/en/technical-documentation/data-sheets/ADXL362.pdf.
- Gyroscope (Figure 1.4.1.5.): An analog gyroscope is used to know the orientation changes of the robot. Devices used is Invensense IXZ-500 dual axis analog gyro. It has a range of -110 to 110 °/s and sensitivity of 9.1 mv/°/s. The device is operated at 3 V DC and to interface it through XADC to the FPGA on port `JXADC`, the output voltage is scaled down to 0 to 1 V using voltage divider circuit with 1 V representing an angular velocity of 110 °/s and 0 V representing -110 °/s. The device measures rotation about x-axis and z-axis. The rotation along x-axis is useful in knowing which direction the robot has turned

whereas the rotation along *z*-axis gives the information on the tilt of robot along the pitch and can be used to prevent the robot from toppling when driving up-slope or down-slope. Link to the device manual: https://store.invensense.com/datasheets/invensense/PS-IXZ-0500B-00-03.pdf.

Figure 1.4.1.5. Gyroscope (figure comes from [6])

- Bluetooth (Figure 1.4.1.6.): The remote control for the robot is any android device with *UofT_SAR_Bot* android app installed. The connection to an android device with the Nexys 4 DDR board is over Bluetooth through module Digilent PmodBT2 connected to port JC on the board. This Pmod has Roving Networks RN-42 Bluetooth device chip on it. The device is operated at a baud rate of 9600 bps. The Pmod communicates with the FPGA using UART communication. Link to the device manual: https://reference.digilentinc.com/_media/pmod:pmod:pmodBT2_rm.pdf.

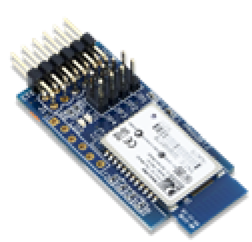


Figure 1.4.1.6. Pmod Bluetooth for Remote Control (figure comes from [7])

- VGA display: During the time of testing of image processing part of the design, it is very useful to have the image from camera displayed before and after processing to evaluate the success of the image processing routines. For this reason, a desktop VGA screen was interfaced through the VGA port available on the Nexys board. This screen is connected only at the time of testing during the project development stage.

1.4.2. Processing and Motion Control

Pixels in the images with an intensity value of blue color are filtered. Then the center of the blob with the largest area is obtained and used as the target. The robot will then move until the target point is located near the center of the image frame.

To avoid obstacles that block the path, the reading from range finders is used by the algorithm in Microblaze to determine if the robot is close to the obstacle. If close enough, Microblaze generates commands to make the robot turn and drive in alternative path in a different direction and then start to rotate back searching for the target to continue driving to the target.

Microblaze runs the algorithms that determine the conditions for driving the robot and sends commands to the motor drivers by generating pulse width modulation (PWM) signals. Possible directions are as follows:

- Forward/Backward: All wheels drive forward/backward at the same speed.
- (Forward/Backward) (left/right): Both left/right wheels drive forward/backward at the same slow speed and right/left wheels drive backward/forward at the same high speed.
- Clockwise/Counterclockwise Rotation: Both right/left wheels drive backward at the same speed and both left/right wheels drive forward at the same speed.

1.4.3. Remote Control

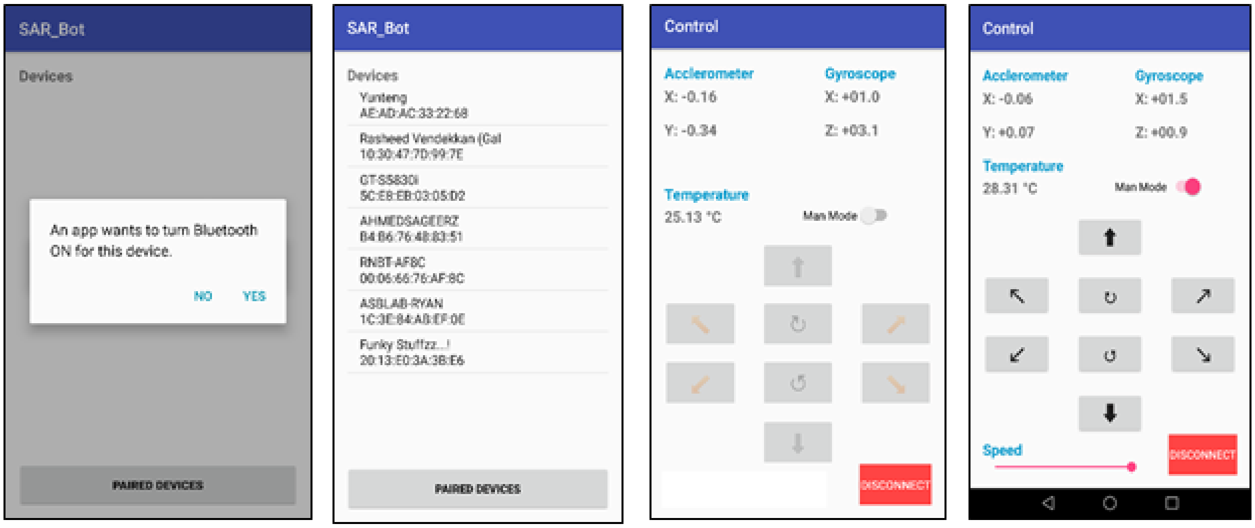


Figure 1.4.3.1. Mobile App GUI (a) connecting page (b) paired device selection (c) control in 'auto' mode (d) control in 'man' mode

The remote-control device is any android device with *UofT_SAR_Bot* android application installed on it. The android app is a java based application built on Android studio app development environment. The app is designed to enable users to control main operations of the robot remotely. The app can be used to connect to turn on the Bluetooth and connect to the paired device. When the app is run, it asks users the permission to turn on Bluetooth if it is off. This is shown in Figure 1.4.3.1. (a). Then in the main GUI page, the app allows the users to select any paired device.

As shown in the Figure 1.4.3.1. (b), RNBT-AF8C is the device name for the Bluetooth module on the robot and it is selected to connect to the robot. This takes to the Control GUI page.

In this page, users can select either 'auto' or 'man' mode of operation using the select switch. When auto mode is selected ('man' mode is turned off), the control page looks like one in Figure 1.4.3.1. (c). The robot in auto mode does automatic target detection and following it with obstacle avoidance. Note that direction buttons are inactive in man mode. If man mode is selected the page looks like as shown in Figure 1.4.3.1. (d) Here the users can the drive robot in any direction by pushing corresponding direction push button, and select the speed of robot motion using the speed slider.

The control GUI page always displays the physical parameters value receive from the robot. This includes: accelerometer value along x-axis and y-axis in unit g, gyro reading along x-axis and z-axis in unit $°/s$, and temperature in °C. Users can switch off the connection using disconnect button and exit if needed.

About the data framing for achieving proper communication between the remote-control device and robot over Bluetooth, the app is programmed to send a specific character (ASCI value) to drive robot in a particular direction. For example, letter `f` is sent to drive robot forward. The Microblaze send the command to the robot to drive forward on receiving the character `f`.

For displaying the accelerometer, gyro and temperature readings, the app has to read the data sent from the board. The data has to be sent in a specific frame structure to identify different values from the stream of data received. In the design, the beginning of a data frame is recognized by the character # and end of a frame is recognized by character ~. The value of accelerometer X follows the start character, followed by accelerometer Y, gyro X, gyro Z and temperature.

2. Outcome

This section summarizes the design outcomes.

The design meets all the functions as planned. The project was completed on time and tested for satisfactory performance. At the conceptual design stage, as detailed in the initial proposal submitted, though the concept of the robot is same as the one designed, there were some components included though they are not required. For instance, the encoder specified in the proposal was found to be unnecessary for achieving the proposed behavior of the robot. Similarly, it was noticed that a four-wheels-driven robot with rough terrain wheels was found to be a better choice than a common two-wheels-differential-driven robot for search and rescue operation robot models.

During the preliminary design stage, it was discovered that the only way to accommodate all the needed sensors without using Pmod expansion was to use analog devices on `JXADC` port. This was a tradeoff between getting Pmod expansion (longer order lead time and cost) and using analog infrared sensors instead of commonly used digital ultrasonic range-finders. Also, at this stage milestone targets were reorganized based on the analysis of the components' availabilities.

In the final stage of the detailed design, additional circuits were also needed to build for voltage dividing. Due to the high power consumption by the four wheels, though the option to use battery as source of power is available in the design, it is powered with external-wired power supply.

Some of the future improvements and suggestions for the project are as follows:

1) Use digital expansion to include wheel encoder enabling the option to know the location of robot by wheel odometry.
2) Use higher rater NiMH battery pack instead of alkaline battery for their recharge ability.
3) Expand the capability of robot to send images of the local environment over Bluetooth to remote device.
4) Include option to allow robot to send audio signals to remote users giving more information about the field environment to the remote user.
5) Use stereo camera to allow finding depth of object in image allowing robot to use dynamic speed setting. For instance, if the target is far, the robot could travel faster.

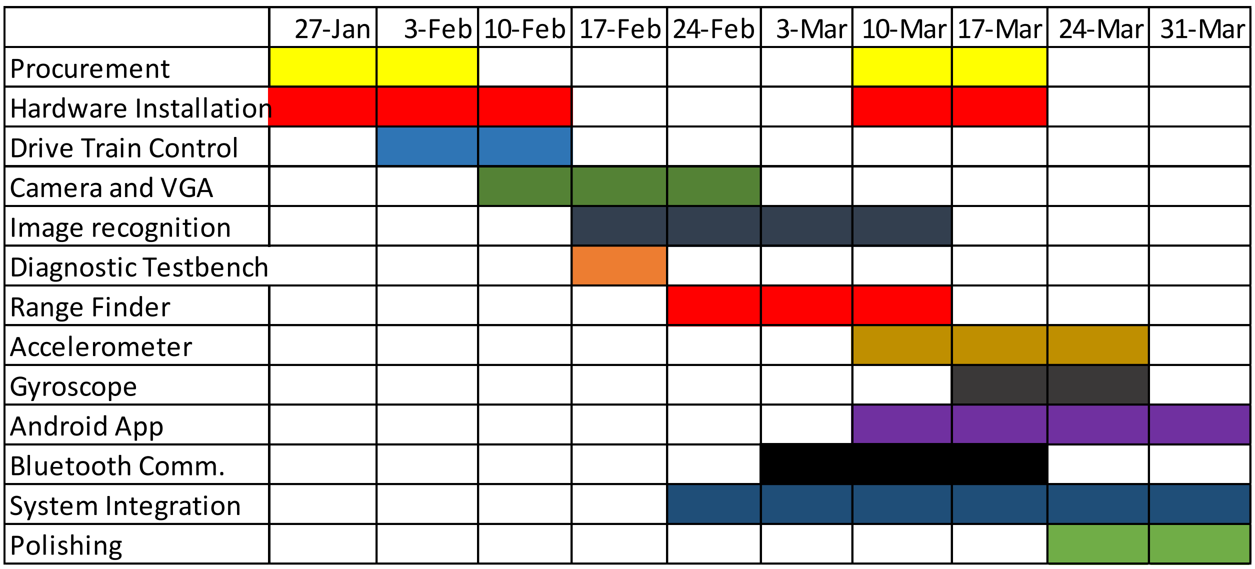
3. Project Schedule

This section gives a comparison between the expected and actual project schedule.

Table 3.1. Expected Project Schedule

Date	Task
10-Feb-17	Motor Interface + Assembling
17-Feb-17	Temperature Interface + Camera Interface (partial)
3-Mar-17	Camera Interface (complete) + Accelerometer Interface + Gyro Interface
10-Mar-17	Range-Finder Interface + Wireless Interface (partial)
17-Mar-17	Wireless Interface (complete) + Image Processing (partial)
24-Mar-17	Image Processing (complete) + Motion Controls

Table 3.2. Actual Project Schedule



The team's time management followed the Gantt chart in the proposal closely but with some deviation. The actual work schedule is depicted in the Gantt chart attached. Most of the parts were acquired at the beginning of the project and they were immediately put together in the first week. At the end of week one, the robot's drive train's hardware is functional and the second week was mostly on fine tune the turning and PWM control. The camera data read in and VGA displayed immediately followed and direct camera to VGA transferred which was achieved in one week. Because all the board's digital IOs were occupied by Pmod devices, so only analog ports were left for gyroscope and infrared range finder. That is reason why there is a major refit and rearrangement for all devices. This process took dragged the workflow back for two weeks.

Usually, getting one device working is not difficult but the general integration process takes a long time. For instance, the Android app development was finished early. However, the data transfer subroutines design took more time compared with expected. Despite these obstacles, the original plan predicted the workflow accurately.

4. Description of Blocks

This section describes the design blocks. The description follows a top-down approach, i.e. it starts with the top-level block design and goes into details on each IP block used.

Figure 4.1. provides a graphical representation of the top-level design.

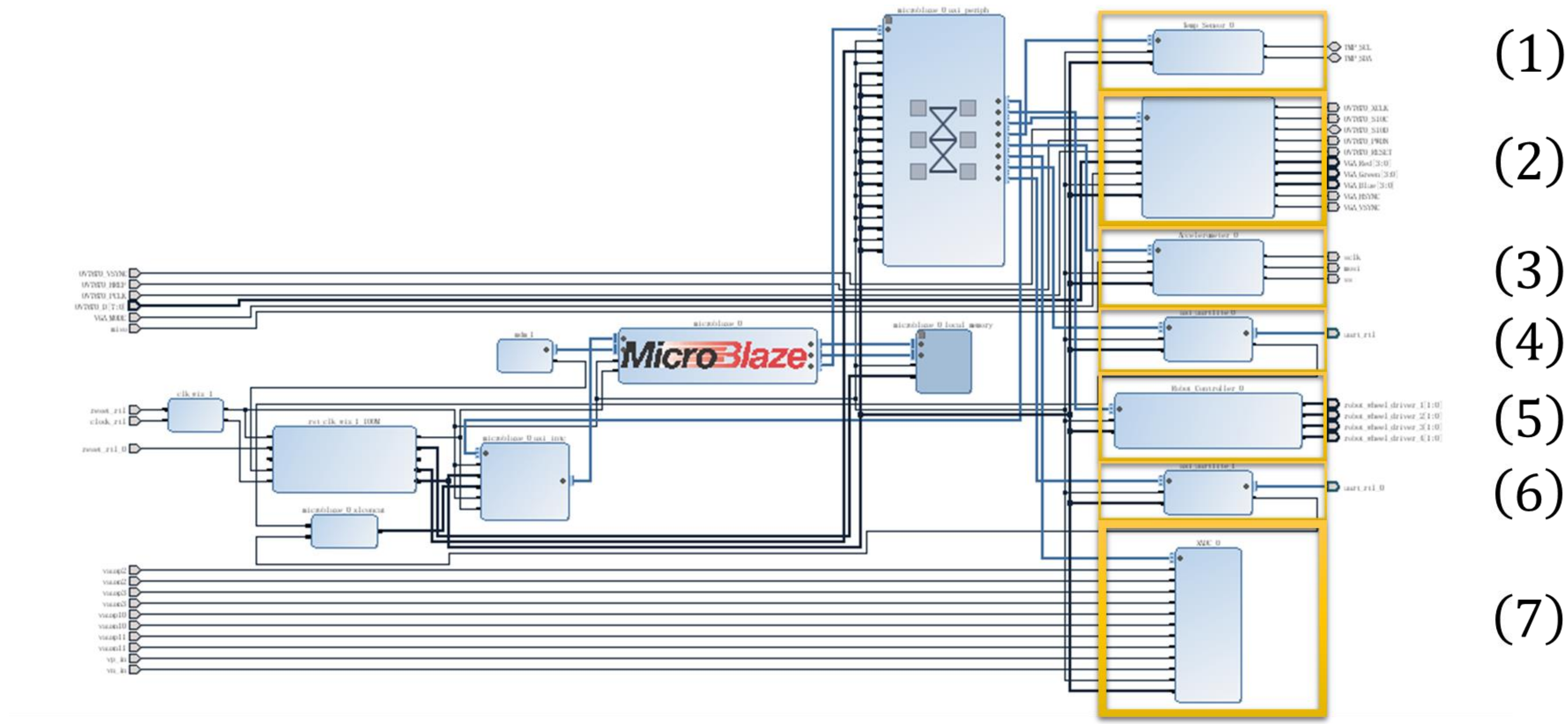


Figure 4.1. Top-Level Block Design

Block Description	**(IP Name)**		
(1) Temperature Sensor	(*Temp_Sensor*)	(4) Terminal UART	(*AXI_UARTLite*)
(2) Image Processing	(*OV7670_VGA*)	(5) Robot Controller	(*Robot_Controller*)
(3) Accelerometer	(*Accelerometer*)	(6) Bluetooth UART	(*AXI_UARTLite*)
		(7) XADC	(*XADC*)

4.1. Temperature Sensor

Part of the code comes from the online Nexy4 DDR demo: https://reference.digilentinc.com/learn/programmable-logic/tutorials/nexys-4-ddr-user-demo/start. The design team modified the original code and made it an `AXI` peripheral that has 1 slave register[1] (Table 4.1.1.).

Table 4.1.1. Slave Register in the *Temp_Sensor* IP block

Slave Register	Description
`slv_reg[0]`	Stores the temperature value in 13 bits: bit 12 is the sign bit; bits 11~4 are the integral part and bits 3~0 are the fractional part[2].

4.2. Image Processing

Part of the code comes from the online PMOD camera and VGA demo shared by teaching assistant Charles Lo: http://www.piazza.com/class_profile/get_resource/ixov7014v1f23d/iyz3c568nc44xz. The design team added several extra modules that filter the captured image pixels (for tracking purpose) and made it an `AXI` peripheral that has 3 slave registers (Table 4.2.1.).

Table 4.2.1. Slave Registers in the *OV7670_VGA* IP block

Slave Register	Description
`slv_reg[0]`	Stores the area of the filtered pixels.
`slv_reg[1]`	Stores the horizontal[3] coordinate of the center of the filtered pixels.
`slv_reg[2]`	Stores the vertical coordinate of the center of the filtered pixels.

4.3. Accelerometer

Part of the code comes from the online Nexy4 DDR demo: https://reference.digilentinc.com/learn/programmable-logic/tutorials/nexys-4-ddr-user-demo/start. The design team modified the original code and made it an `AXI` peripheral that has 2 slave register (Table 4.3.1.).

[1] Xilinx design tool restricts the number of slave registers on an `AXI` peripheral to be greater than 4.
[2] Note that this implies that the temperature value can, in theory, have an accuracy of 0.0625 °C.
[3] "horizontal" (and also "vertical") is from the PMOD camera viewpoint.

Table 4.3.1. Slave Registers in the *Accelerometer* IP block

Slave Register	Description
`slv_reg[0]`	Stores the acceleration measured in the x direction[4] in 12 bits: bit 11 is the sign bit; bits 10~0 are the integral part[5].
`slv_reg[1]`	Stores the acceleration measured in the y direction in the same way.

4.4. Terminal UART

The design team directly used the Xilinx IP block `AXI-UARTLite`.

4.5. Robot Controller

The *Robot_Controller* IP block controls the behavior of the robot. The IP block was made entirely by the design team. It is an AXI peripheral that has 5 slave registers (Table 4.5.1.).

Table 4.5.1. Slave Registers in the *Robot_Controller* IP block

Slave Register	Description
`slv_reg[0]`	Stores the status of the robot: 0 for idle, 1 for active.
`slv_reg[1~4]`	Stores, respectively, the speed of wheel 1, 2, 3, 4 in 32 bits: bit 31 is the direction (1 for forward, 0 for backward); bits 7~0 are the magnitude of the speed.

4.6. Bluetooth UART

The design team directly used the Xilinx IP block `AXI-UARTLite`.

4.7. XADC

Part of the code comes from the online XADC demo: https://reference.digilentinc.com/learn/programmable-logic/tutorials/nexys-4-ddr-xadc-demo/start. Because the Nexy4 DDR board only has one ADC available, the design team added an extra state machine that traverses through all the

[4] x direction is the direction pointing from the back to the front end of the robot, y direction is the direction pointing from the left to the right side the robot, and z direction is the direction pointing upward from the ground. The definitions the same with those in the "XADC" section.

[5] The acceleration is measured in the unit of $0.001g$, where $g \approx 9.8\ \mathrm{m/s^2}$ is the gravitational acceleration.

4 analog ports. Similar to those aforementioned blocks, the design team modified the original code and made it an `AXI` peripheral that has 4 slave registers (Table 4.7.1.).

Table 4.7.1. Slave Registers in the *XADC* IP block

Slave Register	Description
`slv_reg[0]`	Stores the voltage level of the gyrometer `x-out` pin[6].
`slv_reg[1]`	Stores the voltage level of the left range-finder output pin[7].
`slv_reg[2]`	Stores the voltage level of the gyrometer `z-out` pin.
`slv_reg[3]`	Stores the voltage level of the right range-finder output pin.

5. Description of Design Tree

This section describes the structure of the repository that has been posted online at https://github.com/ArmageddonKnight/G9_UofTSARBot.

- `src/` folder contains the source files.
    - `src/Accelerometer` contains the *Accelerometer* IP block.
    - `src/Imge_Processing` contains the *OV7670_VGA* IP block.
    - `src/Robot_Controller` contains the *Robot_Controller* IP block.
    - `src/Temp_Sensor` contains the *Temp_Sensor* IP block.
    - `src/Top_Level_Design` contains the top-level block design.
    - `src/XADC` contains the *XADC* IP block.
- `docs/` folder contains the documentation, which includes the group final report, the slide used at the final presentation, and the project video.

The above descriptions are also included in the `README.md` file in the project repository.

[6] The voltage level of the gyrometer is measured in μV. It is approximately 0.45 V when there is no angular velocity, and there is an increase/decrease of 9 mV when a rotation occurs in the clockwise/counterclockwise direction.
[7] The voltage level of the range-finder is measured in μV. It is 0 V when there is no obstacle at front, and increases linearly to 1 V when the obstacle locates right in front of the range-finder.

6. Tips and Tricks

This section provides design tips for future students. There are two sub-sections. Each sub-section begins with a problem statement and ends with several possible solutions.

The design team would like to provide the following design tips to future students, especially to those who are planning to purchase peripheral devices on their own.

6.1. Voltage Standard

6.1.1. Problem Statement

The V_{DD} (high voltage) of the General-Purpose I/O pins is supplying voltage at 3.3 V. This voltage standard might not apply to all peripheral devices (especially analog devices, many of which are using a voltage different from 3.3 V).

6.1.2. Solutions

Future design teams should first consult shop assistants on the devices operating voltage. Simply purchasing devices that can work at 3.3 V is always a workaround to the issue. Even if they are unable to obtain devices that can work directly under 3.3 V, they still have the option of changing the I/O standard[8] of their output pins in their constraint files. Specifically, they should change the following underlined part to the desired standard:

```
set_property -dict {PACKAGE_PIN * IOSTANDARD LVCMOS33}
[get_ports *]
```

If the design teams are equipped with multimeters (Figure 6.1.2.1.), those meters can be used to verify whether the devices are working or not. For example, if they are working on an analog gyro-meter, they can first connect the V_{DD} and GND pins of the gyro and then connect a voltage detector to the output pin. When rotating the gyro, they should be able to observe the change of the measured voltage.

[8] Table 1-55 of the document *UG471* (https://www.xilinx.com/support/documentation/user_guides/ug471_7Series_SelectIO.pdf) gives a detailed list of supported I/O standards.

Figure 6.1.2.1. Multimeter used by the Design Team (figure comes from [8])

6.2. Frequency

6.2.1. Problem Statement

The clock frequency of the Nexy4 DDR board is 100 MHz. Like the voltage standard, many devices are unable to work at such a high frequency (e.g. the aforementioned *Robot_Controller* block can only work at around 500 Hz).

6.2.2. Solutions

The design teams should start by checking the data sheet and design a clock divider, if necessary, that generates a clock signal which can meet the constraint of the maximum/minimum frequency. If the maximum/minimum values are not provided, they should incorporate the ILA debug core in their design to ensure that the communication protocols have been strictly complied.

7. Acknowledgement

The design team wants to express their sincere gratitude to their project supervisor Fernando Martin del Campo for his kind support throughout the semester, also to Professor Paul Chow for offering the course *ECE532H1 S Digital Systems Design*.

References

[1] "Nexys 4 DDR Reference Manual," Digilent, [Online]. Available: Nexys 4 DDR Reference Manual. [Accessed 12 April 2017].

[2] "DFRobot 4WD Arduino Mobile Platform," RobotShop, [Online]. Available: http://www.robotshop.com/ca/en/dfrobot-4wd-arduino-mobile-platform.html?gclid=CjwKEAjwoLfHBRD_jLW93remyAQSJABIygGpc2Vqp5rS9kl8WXDwdqbvPd4Q3DwLcee0XwooZhuZ3hoCTyfw_wcB. [Accessed 12 April 2017].

[3] "L9110 DUAL MOTOR DRIVER - 0.8A," Creatron, [Online]. Available: https://www.creatroninc.com/product/l9110-dual-motor-driver-08a/?search_query=dual+h&results=160. [Accessed 12 April 2017].

[4] "OV7670 Camera Module," Elecfreaks, [Online]. Available: http://www.elecfreaks.com/store/ov7670-camera-module-p-705.html. [Accessed 12 April 2017].

[5] "SHARP GP2Y0A41SK0F," [Online]. Available: https://www.creatroninc.com/upload/GP2Y0A41SK0F%20Datasheet.pdf. [Accessed 12 April 2017].

[6] "IXZ500 DUAL AXIS GYRO," Creatron, [Online]. Available: https://www.creatroninc.com/product/ixz500-dual-axis-gyro/?search_query=gyroscope&results=11. [Accessed 12 April 2017].

[7] "PmodBT2 Reference Manual," Digilent, [Online]. Available: https://reference.digilentinc.com/pmod/pmod/bt2/ref_manual. [Accessed 12 April 2017].

[8] "Extech DM220: Mini Pocket MultiMeter with Non-Contact Voltage Detector," EXTECH, [Online]. Available: http://www.extech.com/display/?id=14420. [Accessed 10 April 2017].