Development of a Thrifty and Versatile Amphibian Robot using Arduino and RaspberryPi Toolkit

Siddharth Shrotriya

Department of Electronics and Communication Engineering
Jaypee Institute of Information Technology, A-10, Sector-62, Noida, India
siddharth@ieee.org

Abstract— **The field of robotics is extending at an extremely fast rate. To cope with this speed, developers has to design more advanced robots which can interact with the real environment like the natural birds and animals does in their ecosystems. In this paper, the development phases of one such robot are discussed with all the relevant details. This paper is about the development of a multipurpose amphibian robot which can efficiently traverse through water and on the land surface while performing some desired tasks. This development was accomplished using Arduino Mega and RaspberryPi Micro-controller boards which act as the on-site and off-site artificial brain of the amphibian robot. Many different sensors were interfaced with these boards to efficiently interact with the environment. The RaspberryPi which act as an off-site board gives the functionality of semi- controlling the robot and monitoring its state on a screen via wireless communication.**

Keywords— **Amphibian Robots, Communication, Micro-controllers, Robotics Development and Strategic Planning.**

I. Introduction

The development of an amphibian robot is a tedious and meticulous task. Thus, a strategic step by step planning is needed to fulfil such a difficult task. This paper discusses the various phases of development implemented to build a cheap and multipurpose amphibian robot. First, some manual robot models were built to test their functionality and efficiency level in the real environment. Then the materials and other components of the structure of the robot were decided to make it work like a real amphibian animal. Then the final models were designed and the Arduino Mega was installed on it to work like the brain of the robot. All the sensors give their reading to the Arduino [1] micro-controller which analyses the reading and then give required signal to the output devices like actuators and motors. It also gives the signals of its present state to the off-site brain which is RaspberryPi Board installed in the server room situated at a far place via RF wireless communication. The RaspberryPi stimulated the signals and present them graphically on the interfaced LCD screen to give a real world scenario to the user. The user can also give some signals using RF communication to semi-automate the navigation of the robot. This two way communication channel and the multi-functional toolkits of the two Boards provide efficient and versatile working of the amphibian robot.

II. Initial Phase Of The Development

The development of the amphibian robot took place in a number of phases, the first of which is the making of manual prototypes. These models were made using the basic available materials and are driven by using a wired remote. The prototypes shown in Fig. 1 were tested on different types of shores, river beds and flat plains. The key factor in developing the models was the efficient transition from water to land when locomotion take place across the two terrains as it is considered the most difficult work for an amphibian robot. The models were tested on fresh water areas and have to be modified to handle the density of sea water. After making the necessary changes in the weight and type of materials used, the model is set to traverse across like the different amphibian animals present in our natural ecosystem.

TABLE I
FINAL INVENTORY OF THE REQUIRED ITEMS

Purpose	**Selected Item**
Actuator	Rolling lobe-type actuator
Flexator	Fluidic Expansion Drivers
Contractors	Pleated Pneumatic Artificial Muscles
Benders	Monolithic Bending Actuators
Building Material	Epoxy fiber-glass and Epoxy Aramide
Gears	Bevelled and Worm
Internal Connectors	Insulation Displacement Connectors
Motors	Servo and some stepper
Collector	Hand gripping mechanism
Sensors	Touch, Pressure, light, feedback and ranging sensors
Communication	RF Transmitter and Receivers
Automating Boards	Arduino Mega on the Robot and RaspberryPi in the server room

III. Second Phase Of the Development Process

A. Selecting the Right Inventory

The second phase of development of the robot was to select the right components to be used in making of the structure of the robot [2]. The important points to be taken care of in selecting the materials were price of the unit, lightness of the main building material, compactness of inner items of the structure and multipurpose placement of the components [3]. One more important factor is to choose the microcontroller boards which were cheap, contain fast processing capability, fast transition ability between data fetching and addressing. They should have enough memory to store all the commands required to automate or semi-automate the amphibian robot.

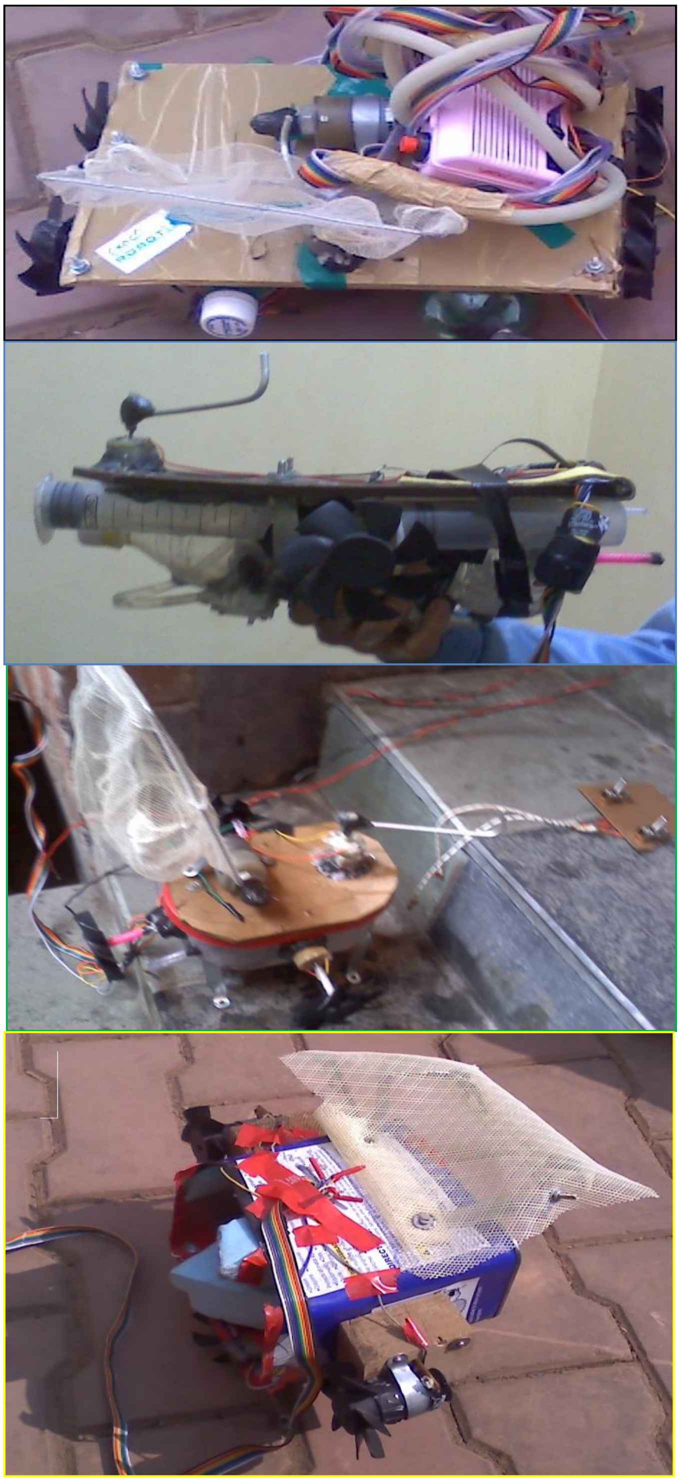

Fig. 1 All the manual models built during initial phase of the development

After taking into consideration the above mentioned points, required components were selected to build the robot. Table 1 shows the final chosen inventory.

```
import pygame.camera, RPi.GPIO as GPIO
import time, signal, Tkinter as tk
from pizypwm import *

pygame.init()
pygame.camera.init()
window = pygame.display.set_mode(352,288),0)
cam = pygame.camera.Camera("/dev/video0",(352,288))
arduino= serial.Serial('/dev/ttyAMA0',9600)
cam.start()
image = cam.get_image()
pygame.image.save(image,'abc.jpg)
cam.stop()
first_f   = PiZyPwm(0.01, 100, 12, GPIO.BOARD)
second_f   = PiZyPwm(0.01, 100, 13, GPIO.BOARD)
# sleep and stop commands
def begin(contrast):
  setup()
  GPIO.output(RST, False)
  time.sleep(0.100)
  GPIO.output(RST, True)
  lcd_cmd(0x21) # extended mode
  lcd_cmd(0x14)
  lcd_cmd(contrast)
  lcd_cmd(0x20)
  lcd_cmd(0xc) # non-inverted display
  cls()
```

Fig. 2 A Part of the python code used to program the RaspberryPi Micro-controller Board

B. *Automating Boards Used in the Project*

There were two boards that were used for the purpose of making the robot autonomous or semi-autonomous. The first one was Arduino Mega Micro-controller Board. This board is based on the ATmega2560. It has 54 digital input/output pins (of which 14 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Mega is compatible with most shields designed for the Arduino Duemilanove. The Mega2560 differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the ATmega16U2 (ATmega8U2 in the revision 1 and revision 2 boards) programmed as a USB-to-serial converter. The second was RaspberryPi Model-A Development Board. The Raspberry Pi is a credit-card-sized single-board computer developed in the UK by the Raspberry Pi Foundation with the intention of stimulating the teaching of basic computer science in schools. The Raspberry Pi has a Broadcom BCM2835 system on a chip, which includes an ARM1176JZF-S 700 MHz processor can be overclocked till 1 GHz, Video-Core IV GPU, and originally shipped with 256 megabytes of RAM, later upgraded to 512MB. It does not include a built-in hard disk or solid-state drive, but uses an SD card for booting and long-term storage. The first board was installed on the robot to operate it with the help of the signals from the installed sensors and the signals received from the server room. The second was used in the server room (as shown in Fig. 5) to display the working of the robot on the interfaced LCD and gives signals to the robot as per the requirement of the user.

```
void setup() {
pinMode(lcPin, INPUT); //listens to the LANC line
pinMode(cdPin, OUTPUT);
pinMode(rcButton, INPUT);
digitalWrite(rcButton, HIGH);
digitalWrite(cdPin, LOW); //set LANC line to +5V
delay(5000);
bitDuration = bitDuration - 8; }

#ifdef TRANSMITTER
void loop() {
  static int prev_button = BUTTON_NOT_PRESSED;
  int cur_button;
  cur_button = digitalRead(BUTTON_PIN);
  if ((prev_button == BUTTON_NOT_PRESSED) &&
  (cur_button == BUTTON_PRESSED))
  {    writeUInt(271);  }
  delay(50); // Debounce button
  prev_button = cur_button;
  D5.shutterNow();
  delay(5000);
  D5.shutterDelayed();
  delay(5000); }
```

Fig. 3 A Part of the Arduino C code used to control the Arduino Mega micro-controller Board

C. Programming the Robot

The programming languages used in this system were python and Arduino C. Python [4] was used to code the RaspberryPi [5] with the help of the Debian Linux OS booted on the SD card which was inserted in its desired slot of the board. Arduino C is a variant of Embedded C which was used to code the Arduino Mega Board which was operating the main robot [6]. Part of the python code is shown in Fig. 2 and part of the Arduino C [7] code is shown Fig. 3. These two codes run simultaneously communicating with each other and with the surrounding environment fetching and transferring the data collected with the help of the ranging and feedback sensors. This data was used to control the locomotion of the robot.

IV. Setting up the Complete Arrangement

This was the last phase of the development. Fig. 4 shows the meticulously modified and finalized model versions of the structure of the robot designed after taking into considerations the working and efficiency of the models shown in Fig. 1. These models were built using the actuators and other material listed in Table 1.

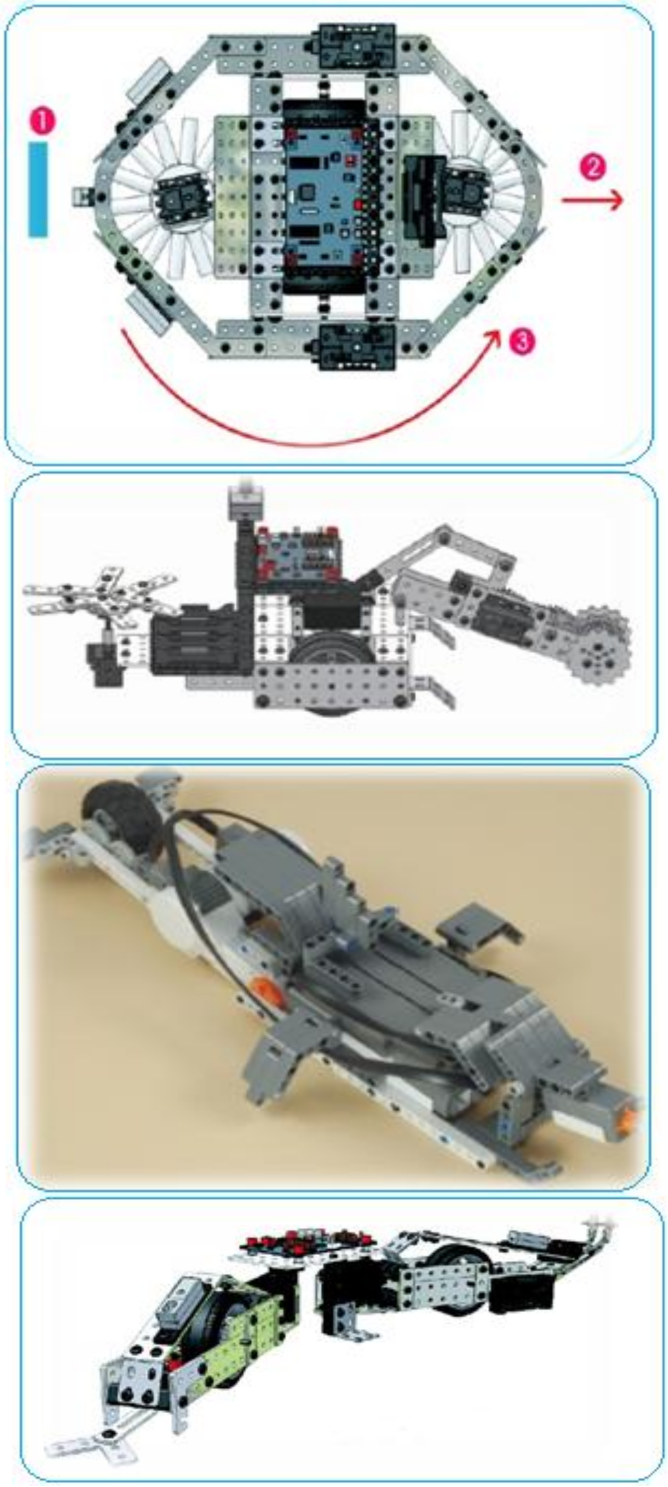

Fig. 4 Final design-parts of the complete model of the Amphibian Robot

The touch sensors were planted on the front, sides and rear part of the robot which senses a haptic response from the surroundings and send a corresponding output to the micro-controller. The pressure sensor were installed on the upper part to sense the water pressure applied on the robot and send the required signals to the micro-controller so that it can make necessary changes to the position of the robot in terms of the depth below the sea level.

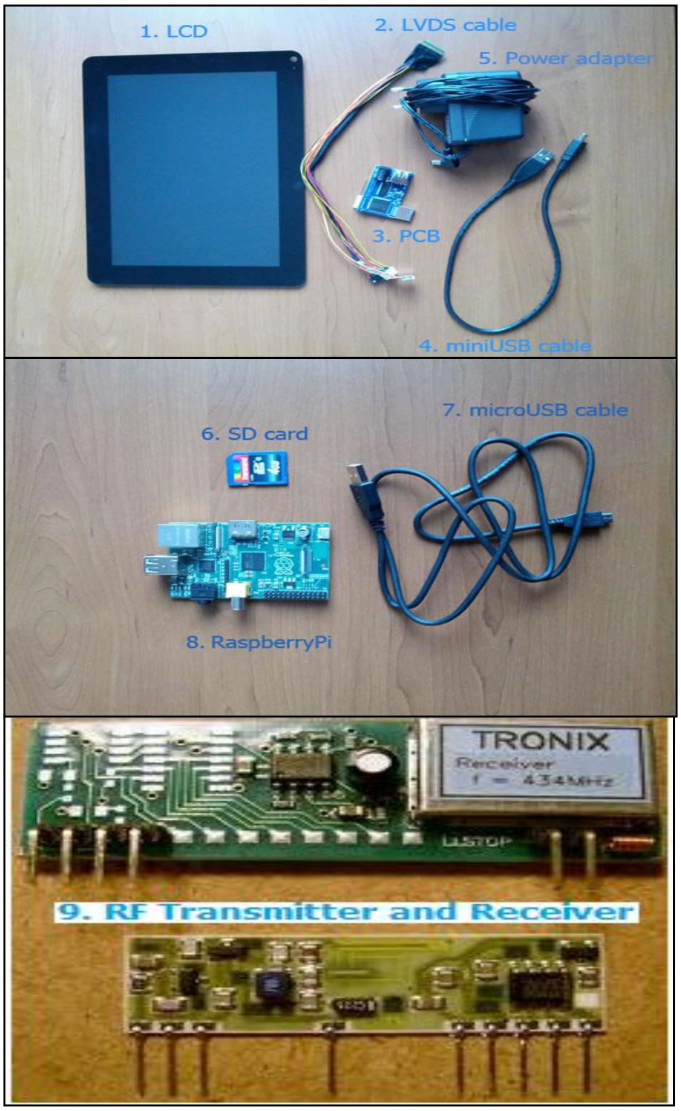

Fig. 5 Display of the components used in the working of the server Room

Light sensor adjusts the front and bump light of the robot according to the surrounding light intensity. Camera [8] implanted on the robot capture the environment images and sends them in array form to the server room. Feedback sensor such as temperature sensor was provided on the robot to notify the server about the environment conditions. Ranging sensors like ultrasonic and IR sensors were used to determine the long and short range obstacle inside the water and on the surface. These sensors help the robot to map its way in water and also to run on plain floors. All the readings from these sensors were fed to Arduino Mega Board which processes it and then automate the robot by giving signals to the actuators and locomotion motors.

The robot communicates with the server room (shown in Fig. 5) using the RF transmitters and receivers installed on both the ends of the communication channel. The artificial brain of the server room was the RaspberryPi Board [9]. This Board was interfaced to an LCD, keyboard, mouse and RF modules. It shows the generated path of the robot moving under the water surface on the LCD screen. It also shows all the signals received from the robot in a graphical form on the screen. Through this so called computer, a user can control some parts of the robotic system and change its navigation routine. This is achieved by sending the signals via RF transmitter to the micro-controller (brain) of the robot. This bi-directional communication along with the finely decided structure helps the robot to act like a real amphibian animal in the coastal environment [10].

V. Conclusion

The complete set-up shown in the paper was successfully built by using the shown operators and attributes of the robot. It efficiently moves in water and on the surface performing some actions specified by the user. In future, we want to develop robots which can efficiently think like a scuba diver while performing all the tasks of a regular amphibian robot.

References

[1] Corey McCall, Branden Maynes, Cliff C.Zhou and Ning J. Zhang, “RMAIS: RFID –based Medication Adherence Intelligence System”, 32nd Annual International Conference of the IEEE EMBS, 2010.
[2] Chunlin Zhou, Ong.T.W. and Junzhi Yu, “Modular design and initial gait of an amphibian robotic turtle”, Robotics and Biometric, 2007, 535-540.
[3] I. Gaiser, R. Weigand, O. Ivlev, A. Andres, H. Breitwieser, S. Schulz and G. Bretthauer, “Compliant Robotics and Automation with Flexible Fluidic Actuators and Inflatable Structures”, Smart Actuation and Sensing Systems – Recent Advances and Future Challenges, 2011, Ch-22.
[4] Mark Lutz, Programming Python (Fourth Edition, O’Rielly Media, 2011)
[5] RaspberryPi: http://paper.li/alastair_hm/1333999192 and the other is: http://en.wikipedia.org/wiki/Raspberry_Pi
[6] A. Crespi, A. Badertscher, A. Guignard, and Ijspeert. A.J. “Swimming and crawling with an amphibious snake robot”. Proceedings of the 2005 IEEE International Conference on Robotics and Automation (ICRA 2005), pages 3035-3039, 2005
[7] Arduino Mega: http://arduino.cc/en/Main/ArduinoBoardMega
[8] Camera Interfacing: http://letsmakerobots.com/node/13354
[9] GPIO of RaspberryPi: http://code.google.com/p/raspberry-gpio-python
[10] Alexander S. Boxerbaum, Philip Werk, Roger D. Quinn, Ravi Vaidyanathan" Design of an Autonomous Amphibious Robot for Surf Zone Operation: Part I Mechanical Design for Multi-Mode Mobility" Proceedings of the 2005 IEEE/ASME International Conference on Advanced Intelligent Mechatronics Monterey, California, USA, 24-28 July, 2005