Presentation Script

1st slide: Title

Good morning, madam/sir, I’m Binari Dissanayake.

In this period many wildlife photographers are facing severe threats while taking photos due to animal attacks. So, we are going to introduce a new device to avoid such kind of problem. It is Automated Camera Stand.

2nd slide: Group Members

These are our group members.

3rd slide: Problem In brief

Let’s go through our problem.

In modern cameras, we have the feature to take remote photos we called it remote photography. Therefore, we can take pictures even we are far from the camera. But there is no way to bring the camera near to the animals rather than a person carry the camera.

· When we analyze the problem, we found 3 main sides of this problem.

· When we take from the photographer's side; Imagine we need to capture a very close wild animal photo, we can't reach the animal. Because it will be very dangerous. In the past few years, many wildlife photographers have died due to animal attacks.

· We have to spend a lot of time, until the required animals come, to capture their photos.

· When the photographer needs to take a good quality photograph, taking a close correct angle is an essential thing. But the photographers can't go near the animals to achieve those requirements. Furthermore, from time to time light conditions is constantly changed.

· From the animal's side: If the animal is an innocent one, it does not stay until we go near them.

· When we get the environmental side: the photographer's behavior caused to change the biodiversity.

This is our problem in brief.

4th slide: Project Aim

Now we come to our project aim. Our aim is Automate the distance photography process to make sure it is safer and effective.

5th slide: Objectives

When we discuss our project objectives, we mainly expect 6 objectives.

Those are ,

· Rotate the camera both vertically and horizontally,
· Move the stand according to given GPS coordinates,
· Reach the target safely,
· If an animal tries to attack the camera holder, then we can sound the siren. Like that we can ensure the safety of the camera.

- Control the camera holder using a remote controller.

6th slide: Proposed solution

Let’s move to proposed solution.

As the input we use Remote Controller, Keypad, GPS Sensors, magnetometer module and four Ultrasonic Sensors. The output devices are gear motors, servo motors, Display and siren.

In these processes,

- The Remote Controller provides the signals related to motions Gear motors and Servo motors. We use radio frequency technology to communicate with remote controller and camera stand. And also, provides signals to sound the siren when the animals come to attack the camera stand.

- By rotating these servo motors,, we can rotate the camera 90 degrees vertically.

- The camera stand can move forward, backward and left and right .

- The keypad gets the GPS coordinates which need to go to a required place. To find the current location we use the GPS sensors. When the camera stand finds the current location and reaches the required place accurately. When we get user GPS coordinates inputs, we can see these coordinates in the LCD Display.

- By using the magnetometer module, we can take the angle from the north which we need to implement GPS location guiding algorithm.

- By using Ultrasonic Sensors, the camera stand can avoid the obstacles and go ahead.

- This is our overall proposed solution.

Thank you, I give my chance to Dasuni Rathanayake.

Thank you Binari

Lets see how this works In the real time scenario

- Capturing the amazing photographs and the behavior of the wild life is the dream of a wildlife photographer.
- So, our invention reveals the path to that amazing work.
- Lets take a scenario like this.
- Okay lets assume that we are seeing a beautiful bird is on a tree branch too much far away and we want to take a photograph of it.
- But practically there may be lots of difficulties to reach that place where the bird is. as an example, bird may be frightened and flies away.
- So that is the best time to get the usage of our invention.
- First we send our device that means the camera stand in auto mode to the desired destination very easily avoiding obstacles.
- we use GPS coordinates for ease of that process. But if the user prefers to send the camera stand manually through the remote controller it is also possible in our device.
- Now it's time to capture wildlife photographs.
- Before that, the user should change the mode to manual mode.
- And the camera should be a remote controlled camera. It is required.
- Then we can change the angle to focus the camera to the bird using the joystick in the remote controller.
- Not only that then let's assume a big animal as an example an elephant comes near to our camera stand and tries to harm it.
- Okay we can avoid that harm by using the siren. The sound of the siren is enough to chase away harmful animals.

Upeksha

I'm Upeksha

Circuit Diagram

In our project, the automated camera stand, mainly there are two parts. They are the Host and the remote controller. Therefore, we designed two circuit diagrams. One for the host and the other one for the remote controller. And also, there is a power circuit too. We use each ATmega32 microcontroller for the host and the remote controller.

Now let's move to the circuit diagram of the remote controller.

Remote

This is the circuit diagram of the remote controller. Here we have used two joysticks to give inputs to ADC pins of the microcontroller. And the LCD display is connected to SDA and SCL pins using I2C to give digital outputs. The nRF transceiver module is connected to SPI pins that facilitate serial data communication. And also, there is one switch connected to configure the mode of auto/manual and the other switch for configure the siren's on and off status. These two switches are connected to Port D. We have supply power for each and every component by using the power supply circuit.

Circuit Diagram (camera stand) - pins

These are the pins of different components which are connected to the atmega32 which is in the remote controller.

Camera Stand

Now let's move to the circuit diagram of the Camera stand. Here we have used another ATmega32 microcontroller to connect the components of the camera stand's circuit. There is a 4x4 keypad that is connected using 8 pins of Port A. And the nRF transceiver has connected to the SS,MISO,MOSI,SCK,INT2 and PB1 pins. And the GPS module has connected to the RXD and TXD which are facilitated for serial data communication. 4 ultrasonic sensors have been connected by using one multiplexer due to the lack of pins. Trigger pins of all the four ultrasonic sensors are connected through a multiplexer to PC6. And all four echo pins are connected from the multiplexer through the INT0 pin.

We use the Magnetometer module to detect the facing direction of the camera stand according to the true North.

We connected the servo motor to the PWM pin, that means the PB3 pin. And also, 5V power supply has been supplied for each of the above components.

And also, the 4 gear motors have connected to the motor driver module and the motor driver module has connected to the PD4, PD5, PD6 and PD7 pins. 12V and 5V are supplied to the motor driver. The siren connected to the PA7 pin and 12V has been supplied for that pin.

Circuit Diagram (camera stand) - pins

These are the pins of different components which are connected to the atmega32 in the camera stand.

I'm Sadini Pathirana,

Let's move to the 3D Animation of the camera stand.

1st animation-this is the infront of the camera stand ,here you can see ultrasonic sensors.the red colour one is the siren.
there is a camera on the camera stand,which is connected to the servo motor.these are lcd display and the keypad.
this is the backside of the camera stand.

2nd Animation-In this video,you can see the functionality of the camera stand,
the wheels are working with the help of the gear motors.the camera changes the vertical angle with help of servo motors.

3rd Animation-In this video,you can see how the internal parts are fixed together.
This is the project box,it has magnetometer module,motor drive,atmega32 ,nRF transceiver and the GPS module

Now you can see how the camera stand avoid the obstacles.
Next you can see how the camera capture the objects by changing the vertical angle.

Resources-Software

These softwares are,
proteus -for circuit simulation
Solidworks and blender -to design the 3D diagram and animation
Microchip studio-Microcontroller coding
Git -GitHub-for version controlling
KiCad - PCB layout designing

Resourse - Hardware

These are the h to use
Microcontroller - ATmega32
GPS Module
nRF transceiver
Ultrasonic sensors
Gear Motors
Servo Motor
LCD display
Motor Driver Module
Magnetometer Module
4x4 Keypad
Li-Po Batteries
Joystick Module
siren

Cost Estimation

Our expected cost estimation is nineteen thousand seven hundred and forty rupees.
These all prices of components are taken by referring Lanka tronics Pvt Ltd and microchip.lk,Tronics.lkwebsite.
It's time to test and implement. I'll hand over this presentation to Upeksha to carry out

Upeksha

Test and Implementation

In our project, there are mainly two components, and one is the remote controller, and that part is done by two members and the other part is the host and that is done by three members.

Ok, let's go to my individual contribution. I'm Upeksha

I'm responsible for the NEO-6M GPS module and the HMC5883L Magnetometer module in our project. Therefore, I have to understand the concepts of USART and I2C protocol. Because the GPS module communicates with the microcontrollers using the USART communication and the magnetometer module communicates with the microcontroller using the I2C protocol.

GPS module

If I go through with the GPS module this uses the USART communication to communicate with the microcontroller. And when thinking about the specifications of the microcontroller, the operating voltage is 3.3V.

Its serial baud rate lies between 4800 bits per second to two lacks thirty thousand four hundred bits per second. But the default serial baud rate is nine thousand six hundred. I used that value for my coding stuff too.

The max supply current for this module is 67 mA.

When taking the pin configuration of the GPS module, there are 4 pins. Vcc pin is for power supply and GND pin is used to ground the GPS module. TXD pin is for the transmission of data and the RXD pin is for the receiving of data. In this project, I have only to transfer data from the GPS module to the Atmega32 chip, and therefore I was only needed to connect the TXD pin of the GPS module with the atmega32 RXD pin. RXD pin is the 14^{th} pin. This is all about the pin configuration of the GPS module.

Internal structure of the GPS module

In the GPS module, there is a GPS antenna to take GPS signals through Radio Frequency which are sent from the GPS satellites. The SAW filter is placed in the GPS antenna. This SAW filter is known as the GPS filter too.

RF front end, short for **radio frequency front end**, is a generic term for all the circuitry between a receiver's antenna and the particular module.

A digital filter is a system that performs mathematical operations on a sampled, discrete-time signal to reduce or enhance certain aspects of that signal.

SRAM or static ram is use to store the gps coordinates or whatever the data temporarily inside the GPS module.

GPS engine provides superior navigation performance under dynamic conditions in areas with limited sky view like urban canyons.

The ARM7TDMI processor implements the ARMv4 instruction set. And this is a 16 bit fast ICE.

The backup RAM is a volatile memory. The backup RAM provides a limited number of bytes of RAM that are retained during the Lightning Protection Measures System.

A real-time clock (RTC) is a battery-powered clock that is included in a microchip in a computer motherboard. This microchip is usually separate from the microprocessor and other chips and is often referred to simply as "the CMOS" .

This is the internal structure of the GPS module.

From the GPS module it will transfer series of characters and when taking GPS coordinates we have to filter the correct characters which we need by using conditional statements with the help of USART communication.

Circuit Diagram

When talking about the circuit diagram. For USART communication we use four pins in the Atmega32 microcontroller. They are RX, TX, XCK and XDIR. Here RX pin is used to take inputs to the Atmega32 in the USART communication and the TX pin is used to transmit data from the Atmega32 microcontroller in the USART communication. XCK pin is used to send the clock signal and the XDIR pin is used to set the direction. When we take the Atmega32 microcontroller RXD pin is the 14^{th} pin. I only needed to take inputs from the GPS module and that's why I only used RXD pin. I connected the GPS module's TX pin to the Atmega RXD pin. Then the GPS module's output will be taken as an input in the microcontroller. That's how this circuit diagram works.

As the inputs for the GPS module, we only supply the power. As the outputs from the GPS module, I take the latitude and longitude values. Because we need the current location's GPS coordinates to build our GPS location guiding algorithm.

Here I have used a 16 by 2 LCD-display module to display the latitude and longitude values that are outputted from the GPS module. That goanna bee uses for testing purposes.

Compiled Code

This is the compiled code that I used for the UART. USART stands for Universal Synchronous Asynchronous Receiver Transmitter. To do the serial communication we use USART technology. Here we do not need the Synchronous data transmission to work with the GPS module. Because we do not need the clock for taking outputs from the GPS module. And therefore, there will be no synchronous data transmission between the GPS module and the microcontroller. That's why we can say that there is a UART communication in here.

For the UART communication, we use 5 basic registers. They are the UDR register, UCSRA register, UCSRB register, UCSRC register, and the UBRR register. Here I have used two macros to define the USART board rate and the BAUD_PRESCALE. Here first, we have to initialize the UART, it's just like that we make the path between the GPS module and the microcontroller. For that, I wrote the function UART_init. The data are passed like characters in this communication method. And therefore, to receive a character from the USART Dara Register I wrote a function called UART_RxChar, and to transmit a character wrote a function called UART_TxChar. To send a whole string through UART I wrote the function UART_sendString.

Magnetometer module

For the GPS location guiding algorithm, we need to identify the current facing direction of the camera stand. For that, I'm using an HMC5883L magnetometer module. This uses the I2C protocol for communication with the Atmega32 microcontroller.

According to the documentation, the supply voltage for the magnetometer module is 3.6V. The frequency to pass serial data is up to 400kHz and the operating temperature is between -22 to 185 Fahrenheit. And the supply current is 2 micro Ampiers. And the power consumed by this module is 360 micro wot.

Magnetometer module pin configuration

When we think about the pin configuration there are 5 pins in the magnetometer module. The Vcc pin takes power to the module, and the GND pin is to ground. SCL is the serial clock pin, and the SDA is the serial data pin. DRDY is the data-ready interrupt pin. But I only needed the SCL and SDA pins to connect with the I2C buss.

Internal Structure Magnetometer module (HMC5883L)

Earth's magnetic field is present in space which points towards the magnetic north. Current carrying conductor also generates a magnetic field around itself. Hence, whenever a current carrying conductor is placed in space, it experiences the effect of the earth's magnetic field affecting the flow of the electrons through that conductor. These changes in the flow of the electrons are used for identifying the heading or direction of the magnetic field. This is the basic working principle of the magnetometer.

- **HMC5883L** uses a magnetoresistive sensor arranged in a bridge circuit, which is made of nickel-iron (Ni-Fe magnetic film) material.
- Its electrical resistance varies with the change in the applied magnetic field.
- The correspondent movement of the nickel-iron material in space experiences earth's magnetic field which changes the material's resistance, and hence we get resultant voltage changes across the bridge. This change in voltages is used to get the magnetic field direction in space.

Circuit Diagram

The magnetometer module uses the I2C protocol to communicate with microcontrollers. That is also a serial communication protocol. For I2C the SCL and the SDA pins have to be used. SCL pin is the 22^{nd} pin in the Atmega32 microcontroller and SDA pin is the 23^{rd} pin in the Atmega32 chip.

I’ll be no need to connect the DRDY pin of the magnetometer module to the Atmega32 microcontroller. We can connect up to 255 components to this I2C buss in this Atmega32 microcontroller. And therefore not only the magnetometer module but also we had to connect the LCD display also to theis I2C buss.

I was not able to find the proteus library for the HMC5883L magnetometer module and I created one using the proteus simulation software. This can’t be used for simulation purposes. This is just for demonstration purposes.

And therefore in the final simulation we will be using an atmega32 microcontroller instead of this magnetometer module.

Binari

In our project, my responsibilities are getting inputs related to the angle of the servo motor, transmitting the processed data from the microcontroller, and avoiding obstacles using 4 ultrasonic sensors .

Also, Dasuni and I are responsible for all the remote controller functions. The microcontroller of the remote gets analog inputs related to the angle using a thumb joystick. Furthermore, I used a nRF24L01 Transceiver module as a transmitter to transmit the above-processed data from the microcontroller. So in our project I had to use thumb Joystick module, nRF24L01 Transceiver module and HC-SR04 Ultrasonic Sensor.

Thumb Joystick

Now let's talk about the used components by me one by one. I used a thumb joystick module to get input to set the vertical angle of the camera. We can move the camera up and down using this joystick. The joystick uses ADC to communicate with the microcontroller.

The joystick is a device that translates our hand movement into an electronic signal, and the movements are converted by the computational unit into entire mathematics, in other words, the joystick translates entirely physical movements.

When we talk about the specification of the thumb joystick, we want 5V to operate the joystick. The operating current is 3.5 mA. In the joystick module, internal potentiometer values are 10kΩ. The operating temperature of this joystick module is 0 to 70 C.

There are 5 pins in the joystick module. The first one is the GND pin. We connect this pin to the ground. We supply power to the module using the VCC pin. The third pin is the VRX pin. Using this pin, we can get a readout of the joystick in the horizontal direction, that means the input related to the X-axis. The fourth pin is the VRY pin, using this pin, we can take a readout of the joystick in the vertical direction, that means the input related to the Y-axis. The last pin is the SW pin. This is the output of the internally connected push button. It is normally open; it means that the digital readout from the SW pin will be high. When the button is pushed, it will connect to the ground by giving output as low.

Internal Structure

When we talk about the internal structure of the joystick module, usually, the joystick module gives values along the x-axis and y-axis. So, it contains two potentiometers, one for each axis. As we move the joystick, the value of the resistance of both potentiometers get change. The potentiometers are connected between +VCC and Ground. They simply behave as voltage divider network. The joystick has one freewheeling Holder. According to the holder movement, the potentiometer knob changes its position and resistance of the potentiometer.

This module produces an output of around 2.5V from X and Y when it is in resting position. Moving the joystick will cause the output to vary from 0v to 5V depending on its direction. If we connect this module to a microcontroller, we can expect to read a value of around 512 in its resting position. When we move the joystick, we should see the values change from 0 to 1023 depending on its position.

Circuit Diagram

Now let's come to the circuit diagram of the joystick module. I created the circuit diagram using the proteus simulation software. In proteus, there is no joystick module. Also, I couldn't find any library for the joystick module. So, I used a variable resistor to represent an axis. I connected this variable resistor to ADC pin which is PA0 pin in the microcontroller. Because I get an Analog reading using this variable resistor. Also, I learnt about ADC technology to communicate.

When using ADC, we can choose ARef or AVCC or Internal as the reference voltage. I selected AVCC as my reference voltage and give 5V to it.

Compiled Code

This is the compiled code of the joystick module. I have written two functions in this, which are initializing ADC and getting readouts ADC. In the main function, I have written the code to get the input of channel A0.

nRF24L01 Tranceiver Module

Now let's come to my next component. I use nRF24L01 transceiver module to communicate with remote controller and the camera stand. Using nRF24L01 module I transmit all controlling instructions to the camera stand. This module uses radio frequencies for that.

The nRF24L01 transceiver module is designed to operate in 2.4 GHz worldwide ISM frequency band and uses GFSK modulation that means gaussian frequency shift keying modulation for data transmission. The data transfer rate can be one of 1Mbps and 2Mbps. And also this module has both transmitter and receiver components.

The operating voltage of the module is from 1.9 to 3.6V, and we supply 3.3 V to the module. Due to logic pins are 5-volt tolerant, so we can easily connect it to any 5V logic microcontroller without using any logic level converter.

The module supports programmable output power viz. 0 dBm, -6 dBm, -12 dBm or -18 dBm and consumes unbelievably around 12 mA during transmission at 0 dBm, which is even lower than a single LED. And best of all, it consumes 26 μA in standby mode and 900 nA at power down mode. Operating temperature is -40 to 85 celsius. In transmitter mode I supply 11.3 mA current to the module. It is important to never to keep the nrf is never in TX mode longer than 4ms.

We can communicate over a distance of 2 km using this nRF24L01 module.

Pin Configuration

Now let's move to pin configuration of nRF24L01 transceiver module. The CE pin is an active-HIGH pin. When selected the nRF24L01 will either transmit or receive, depending upon which mode it is currently in. MISO that means Master in slave out pin is digital output pin. It is used to get SPI output from the nRF24L01. MOSI that means master out slave in pin is a digital input pin which is used to give SPI input to the nRF24L01. IRQ pin is an interrupt pin that can alert the master when new data is available to process. GND pin is the Ground Pin. It is usually marked by encasing the pin in a square so it can be used as a reference for identifying the other pins. CSN pin is an active-LOW pin and is normally kept HIGH. When this pin goes low, the nRF24L01 begins listening on its SPI port for data and processes it accordingly. VCC pin supplies power for the module. This can be anywhere from 1.9 to 3.9 volts. We connect it to 3.3V output from our atmega 32 microcontroller.

Internal Structure

Now let’s come to the internal structure of the nRF24L01 transceiver module. The main thing is this module has the embedded baseband protocol engine (Enhanced ShockBurst) is based on packet communication. It supports various modes from manual operation to advanced autonomous protocol operation. When we see this block diagram there is an Internal FIFOS to ensure a smooth data flow between the radio front end and the system's MCU. Enhanced Shock Burst reduces system cost by handling all the high speed link layer operations.

The RF synthesizer generates a wide range of high frequencies from a single, typically lower, reference frequency with an internal PLL. PLL means Phase Locked Loop, which is a control system that generates an output signal whose phase is related to the phase of an input signal. The output frequency is controlled by accessing the digital registers in the device through an SPI interface.

The nRF24L01 transceiver module communicates over a 4-pin Serial Peripheral Interface (SPI) with a maximum data rate of 10Mbps. All the parameters such as frequency channel (125 selectable channels), output power and data rate can be configured through SPI interface. The SPI bus uses a concept of a Master and Slave, in our project Atmega32 is the Master and the nRF24L01 transceiver module is the Slave.

Okay, You can see there is PA in this diagram. The PA stands for Power Amplifier. It merely boosts the power of the signal being transmitted from the nRF24L01 chip. Whereas, LNA stands for Low-Noise Amplifier. The function of the LNA is to take the extremely weak and uncertain signal from the antenna (usually on the order of microvolts or under -100 dBm) and amplify it to a more useful level (usually about 0.5 to 1V)

The low-noise amplifier (LNA) of the receive path and the power amplifier (PA) of the transmit path connect to the antenna via a duplexer, which separates the two signals and prevents the relatively powerful PA output from overloading the sensitive LNA input. This is the overall idea about that internal structure of nRF24L01 transceiver module.

Circuit Diagram of nRF24L01

Now let's come to the circuit diagram of nRF24L01 transceiver module. I used PB1, PB2, PB4, PB5, PB6 and PB7 pins to connect nrf module and the atmega32 chip.

Compiled code

This is the compiled code for the nRF 24L01 transceiver module.

Ultrasonic sensor

When the camera stand reaches the GPS coordinate automatically, to understand the obstacles and to avoid them there are 4 ultrasonic sensors around the camera stand. Each ultrasonic sensor works on 5V. Trig pin needs 10 microsecond pulse to send the ultrasonic pulse. And the echo pin’s value will be high when the ultrasonic pulse comes back after reflecting from the obstacle. The distance is calculated by using the time taken to travel the ultrasonic pulse. This sensors workind with 15 mA current. Working Frequency is 40Hz. We can avoid obstacles within the range of 2 cm and 4m.

Internal structure

There are two main parts in the ultrasonic sensor viz. transmitter and receiver.

• The transmitter part converts electrical energy into sound and transmits it.

• The receiver part receives the echo and turn this received sound waves into electrical energy.

• This returned echo is measured and used for distance calculation by the ultrasonic sensor. Basically this sensor calculates time interval between signal transmission and reception of echo and determines the distance of the object from the sensor. As this sensor is used for distance measurement it is known as distance sensor.

• Piezoelectric crystals are used in the ultrasonic sensor construction due to the fact that these crystals oscillate at higher range of frequencies.

Ultrasonic sensor Circuit Diagram

This is the circuit diagram of the ultrasonic sensors. We use 1 multiplexers to connect ultrasonic sensors to the microcontroller. All trig pins are connected to pin A4 and all echo pins are connected to pin D7.

Compiled code

This is the compiled code for the ultrasonic sensor.

Dasuni

In our project Binari and I are responsible for all the remote controller functions.

And I am responsible for the Thumb Joystick module, nRF24L01 Transceiver module as receiver and 4x4 keypad. I use ADC Technology and SPI concepts. I have to give inputs to the ATmega32 microcontroller of the remote through the joystick and it uses the ADC Technology for that. Moreover, it gives inputs to the microcontroller of the camera stand from the nRF24L01 Transceiver of the remote controller. nRF24L01 Transceiver uses the SPI to communicate with the Atmega32 microcontroller of the host.4x4 keypad is mainly used to get the GPS coordinates from the user. If necessary, it is used to get some configuration from the user.

Thumb Joystick

In our project I use this joystick to give analog inputs to the ATmega32 microcontroller related to the motions such as moving forward backward and moving left right by using ADC technology.

The joystick is a device that translates our hand movement into an electronic signal, and the movements are converted by the computational unit into entire mathematics, in other words, the joystick translates entirely physical movements.

When we talk about the **specification** of the thumb joystick, we want 5V to operate the joystick. The operating current is 3.5 mA. In the joystick module, internal potentiometer values are 10kΩ. The operating temperature of this joystick module is 0 to 70 C.

There are 5 pins in the joystick module.

1. GND: ground

2. Vcc: 5V DC

3. VRx: voltage proportional to x position

4. VRy: voltage proportional to y position

5. SW: switch pushbutton

Pin 1, 5 - VCC and GND

Supply voltage(+5V) and ground given to Joystick.

Pin 2 –VRx

This pin provides an analog output voltage from 0 volts to VCC according to the movement of Holder in X-direction (axis).

Pin 3 - VRy

This pin provides an analog output voltage from 0 volts to VCC according to the movement of Holder in Y- direction (axis).

Pin 4 - Switch

This pin has one tactile switch.

When a switch is not pressed, this pin is connected to VCC through a resistor.

When a switch is pressed, this pin is connected to Ground.

Internal Structure

Joystick is an input device. Analog joystick is sometimes called as Control Stick. It is used to control the pointer movement in 2-dimension axis.

It is made by mounting two potentiometers at a 90 degrees angle to read user's input.

The potentiometers are connected to a short stick centred by springs. One potentiometer is used to get the analog output voltage for X-Direction movement related to move the camera stand forward backward and the other potentiometer is used to get the analog output voltage for Y-Direction movement related to move the camera stand left right.

The potentiometers are connected between +VCC and Ground. They simply behave as voltage divider network

The joystick has one freewheeling Holder. According to the holder movement, the potentiometer knob changes its position and resistance of the potentiometer.

This module produces an output of around 2.5V from X and Y when it is in resting position. Moving the joystick will cause the output to vary from 0v to 5V depending on its direction. If we connect this module to a microcontroller, we can expect to read a value of around 512 in its resting position. When we move the joystick, we should see the values change from 0 to 1023 depending on its position.

Circuit Diagram

I created this circuit diagram in Proteus simulation software to give analog signals to the ATmega32 microcontroller by using a joystick related to the motions. There is no joystick library in Proteus software . Therefore I used 2 High granularity interactive potentiometer (Pot-HG) instead of joystick.

for my circuit Diagram. One is for forward/backward movements and the other one is for left/right movements. I connected two potentiometers to ADC pins of the atmega32. They are PA1 39^{th} pin and PA2 38^{th} pin. And I give 5V to AReff or AVcc pin of the the microcontroller and the joystick circuit.

Compiled Code

I used microchip studio for the coding part. I have written two functions to initiate the ADC channels and to read the ADC . In the main function, I have written the code to get the input of channel A0.

nRF24L01 Transceiver Module

- Now let's come to my next component. I use nRF24L01 transceiver module to communicate with remote controller and the camera stand.
- Using nRF24L01 module I receive all controlling instructions from the remote controller. This module uses radio frequencies for that.
- The nRF24L01 transceiver module is designed to operate in 2.4 GHz worldwide ISM frequency band and uses GFSK modulation that means gaussian frequency shift keying modulation for data transmission. The data transfer rate can be one of 1Mbps and 2Mbps.
- The operating voltage of the module is from 1.9 to 3.6V, and we supply 3.3 V to the module. Due to logic pins are 5-volt tolerant, so we can easily connect it to any 5V logic microcontroller without using any logic level converter.
- The module supports programmable output power viz. 0 dBm, -6 dBm, -12 dBm or -18 dBm and consumes unbelievably around 12 mA during transmission at 0 dBm, which is even lower than a single LED. And best of all, it consumes 26 μA in standby mode and 900 nA at power down mode.

Specifications :

- Supply Voltage : 3.3 V
- Operating current : 26 μA Stand by mode, 900nA power down mode
- Operating Temperature : -40 to +85 ºC
- Power : 44.55 mW
- Maximum current : 13.5 mA at 2 Mbps
- Fast AGC for improved dynamic range
- Sensitivity : -82 dBm at 2Mbps

Now let’s move to pin configuration of nRF24L01 transceiver module. The CE pin is an active-HIGH pin. When selected the nRF24L01 will either transmit or receive, depending upon which mode it is currently in. MISO that means Master in slave out pin is digital output pin. It is used to get SPI output from the nRF24L01. MOSI that means master out slave in pin is a digital input pin which is used to give SPI input to the nRF24L01. IRQ pin is an interrupt pin that can alert the master when new data is available to process. GND pin is the Ground Pin. It is usually marked by encasing the pin in a square so it can be used as a reference for identifying the other pins. CSN pin is an active-LOW pin and is normally kept HIGH. When this pin goes low, the nRF24L01 begins listening on its SPI port for data and processes it accordingly. VCC pin supplies power for the module. This can be anywhere from 1.9 to 3.9 volts. We connect it to 3.3V output from our atmega 32 microcontroller.

Internal Structure

Now let’s come to the internal structure of the nRF24L01 transceiver module. The main thing is this module has the embedded baseband protocol engine (Enhanced ShockBurst) is based on packet communication. It supports various modes from manual operation to advanced autonomous protocol operation. When we see this block diagram there is an Internal FIFOS to ensure a smooth data flow between the radio front end and the system's MCU. Enhanced Shock Burst reduces system cost by handling all the high speed link layer operations.

The RF synthesizer generates a wide range of high frequencies from a single, typically lower, reference frequency with an internal PLL. PLL means Phase Locked Loop, which is a control system that generates an output signal whose phase is related to the phase of an input signal. The output frequency is controlled by accessing the digital registers in the device through an SPI interface.

The nRF24L01 transceiver module communicates over a 4-pin Serial Peripheral Interface (SPI) with a maximum data rate of 10Mbps. All the parameters such as frequency channel (125 selectable channels), output power and data rate can be configured through SPI interface. The SPI bus uses a concept of a Master and Slave, in our project Atmega32 is the Master and the nRF24L01 transceiver module is the Slave.

Okay, You can see there is PA in this diagram. The PA stands for Power Amplifier. It merely boosts the power of the signal being transmitted from the nRF24L01 chip. Whereas, LNA stands for Low-Noise Amplifier. The function of the LNA is to take the extremely weak and uncertain signal from the antenna (usually on the order of microvolts or under -100 dBm) and amplify it to a more useful level (usually about 0.5 to 1V)

The low-noise amplifier (LNA) of the receive path and the power amplifier (PA) of the transmit path connect to the antenna via a duplexer, which separates the two signals and prevents the relatively powerful PA output from overloading the sensitive LNA input. This is the overall idea about that internal structure of nRF24L01 transceiver module.

Circuit Diagram

I created this circuit diagram in Proteus simulation software to communicate with two ATmega32 microcontrollers of the remote controller and the camera stand by using a nRF24L01 Transceiver module.

There are CE,CSN,SCK,MOSI,MISO,IRQ pins in the nRF24L01 transceiver. And I connected the CE pin to PB1(2),CSN to PB4(4),SCK to PB7(7), MOSI to PB5(5), MISO to PB6(6) and IRQ to PB2(3) pin of the Atmega 32 microcontroller.

Compiled code

I used microchip studio for the coding part and This is the code which we are going to use for the nRF24L01 transceiver module.

4x4 Keypad

4x4 keypad is mainly used to get the GPS coordinates from the user. If necessary, it is used to get some configuration from the user. There are 16 push buttons divided into rows and the columns in it. The first four input lines have been connected to the four rows and the second four lines to the columns.

- Interface: 8-pin access to 4×4 matrix
- Maximum Rating: 5 V, 12.5 mA
- Power : 62.5 mW
- Interface: 8-pin access to 4x4 matrix
- Operating temperature: 32 to 122 °F (0 to 50°C)
- Dimensions: Keypad, 2.7 x 3.0 in (6.9 x 7.6 cm)
- Cable: 0.78 x 3.5 in (2.0 x 8.8 cm)

Internal Structure

This 4x4 matrix keypad has 16 built-in push button contacts connected to row and column lines. All these membrane switches are connected to each other with conductive trace underneath the pad forming a matrix of 4×4 grid. Normally there is no connection between rows and columns.

A microcontroller can scan these lines for a button-pressed state. In the keypad library, the Propeller sets all the column lines to input, and all the row lines to input. Then, it picks a row and sets it high. After that, it checks the column lines one at a time. If the column connection stays low, the button on the row has not been pressed. If it goes high, the microcontroller knows which row (the one it set high), and which column, (the one that was detected high when checked).

Circuit Diagram

I created this circuit diagram in Proteus simulation software to get the GPS coordinates from the user and If necessary, it is used to get some configuration from the user. I connected the keypad and the Atmega 32 microcontroller using PA0(40), PA2(39), PA3(38), PA4(37), PA5(36),PA6(35),PA7(34) pins of the Atmega32.

Compiled code

I used microchip studio for the coding part and This is the code which we are going to use for the 4x4 keypad module.

Sadini

In this project, I am responsible for the servo motor and the siren. I have to change the vertical angle of the camera when taking a photo of a wild animal, by using a servo motor. And also, have to sound a siren according to the user inputs.

Servo Motor

A servo motor is an electromechanical device that produces torque and the velocity based on the supplied current and voltage. A servo motor work as part of a closed loop system providing torque and velocity as command from a servo controller utilizing a feedback device to close the loop. The feedback device supplies information such as current, velocity or position to the servo controller, which adjusts the motor action depending on the command parameters.

In our project, Servo motors is used for changing the vertical angle of the camera within 90 degrees. When we take a photo of a wild animal, we have to change the direction of the camera according to the direction of the animal.

When the signal is received to the Atmega32 microcontroller in the host, it provides a signal to make the correct angle of the servo motor. For that we use the PWM method to process that.

It has three colour wires. the red colour wire should be connected to the power supply. Brown colour wire connects to the Ground and the Orange colour wire connects to the PWM signal given by the Atmega32 in the host.

When we take the specification of the servo motor, weight is 55 g , operating voltage is 5V,operating maximum current is 900mA and the power is 4.5W,torque is 2.5kg/cm.

Internal structure

Inside the servo motor, a small DC motor, potentiometer(If the potentiometer analog value is lower than a threshold, rotate servo motor back to 0 degree. If the potentiometer output voltage is greater than a threshold, rotate servo motor to 90 degree), and a control circuit. The motor is attached by gears to the control wheel. As the motor rotates, the resistance of the potentiometer is change, so the control circuit can precisely regulate how much movement there is and in which direction.

When the shaft of the motor is at the desired position, power supplied to the motor is stopped. If not, the motor is turned in the appropriate direction. The desired position is sent via electrical pulses through the signal wire. The motor's speed is proportional to the difference between its actual position and desired position. So if the motor is near the desired position, it will turn slowly, otherwise it will turn fast. This is called proportional control.

circuit diagram

I have used the proteus software to simulate the circuit, for the testing part, I selected the PB3 PWM pin to connect the servo motor to Atmega32 microcontroller.

Code

The coding part has been done in the Atmel studio software and according to the given angle the servo is rotated degree to degree.

Siren

the siren is used to protect the camera stand. When the wild animals come to attack the camera stand, we just sound the siren to chase away the animals.

When we consider the specifications, operating maximum voltage is 12V,

The operating maximum current is 1.5A, Sound intensity is 112 -116 decibels and speed is 1000 rpm and power is 18W.When the user switch on the siren button to sound the siren, a digital signal is passed from the remote controller to siren .

Internal structures

When we take the internal structure,there is a fan and a motor can be seen inside the siren. the fan is connected to the motor.when the motor is rotating ,the sound is generated.

circuit diagram

I selected the PB0 pin to connect the siren to the ATmega32 through the relay. As it is not enough 5V for the siren, I have used an external power resource to provide the required voltage 12V(10A,DC 30v-300w). when connecting the relay to the Atmega32 microcontroller, a transistor is used between them as a switch that causes high current . When the high current flows through the electromagnet ,the current also goes through the diode and it helps control the current. It causes a magnetic field that attracts the switch and the siren is switched on and sounds the siren.

Codes

For the coding part, I used Atmel studio software and have connected the siren to PB0 pin.If the siren gets input 1,siren sounds.

Hansa’s Individual Contribution

In our project, I am mainly responsible for motor control activities. So I have learnt PWM to control motor speed. There are some other components like displays and buttons that I am responsible for. Let's talk about my components one by one. First,

PWM

Atmega32 microcontroller has 3 timers. Timer0 and timer2 are 8 bit timers and timer1 is 16 bit one. Timer0 is used to servo motors, Timer1 is used to gear motors.

PWM Code

This is the code for PWM

Gear Motors

Our design is a little bit heavy one and the speed should be somewhat low. So we have used 4 180 RPM High torque gear motors. These motors can carry even a huge DSLR Camera. We are going to use these shock absorption wheels to shoot steady shots. When the motors run in maximum torque each motor draws up to 20A in 12V. So we have to use a high discharge li-po battery with 4 cell 35c. The rated voltage of this battery is 14.8V but this will charge up to 16V.

Internal Structure of Gear motors

This is the Internal structure of the gear motor which I have used in our project. There are mainly dc motor part, gear sets.

These are the dimensions of the gear motors.

Circuit Diagram

Code

Motor Driver

Our first plan is to use an L298N motor drive for this project. This drive can handle up to 50V and a maximum of 2.5A. Maximum logical supply and enable voltages are 7V. This is a dual full-bridge driver. So both sides can be handled by a single drive. Pin configuration is very simple. There are 2 input pins, 2 output pins and a single enable pin for each side. For any reason, if this draining current is not enough, as in the second plan, we will use a more powerful motor drive. It is a BTS7960 43A Motor Driver. If so we have to use a couple of these drivers for the project.

Internal Structure of motor driver

Motor Driver Circuit Diagram

The PWM pins are not enough for all 4 motors. Therefore, we have connected the two left-side motors and two right-side motors separately as duals. Then we are doing the work which is done by using 8 pins; by using just 4 pins. But we cannot use all 4 PWM pins in the Atmega32 microcontroller. So we use one PWM pin and 1 digital pin for each side. From that, we can save 2 PWM pins for the rest of the tasks.

Motor Driver Code

This is the code that I wrote for the motor driving task. There are 2 separate functions for each motor and one function to get the motor speeds of both motors.

LCD Display

We use 2 "16 by 2" LCD displays for the remote controller and the camera holder. It is working between 3.3V to 5V and 1.5mA to 2.5mA.
We use this display to give user instructions and current status of the system.

Internal Structure

This is how internal structures are connected.

PCF8574A Remote 8-Bit I/O Expander to I2C

The display uses the 4-bit configuration to communicate with the microcontroller. We had to use SPI pins for the nRF module. So I can’t connect the LCD display and nRF module together. Due to that reason I connect the LCD display via the I2C protocol.
I use have to use 8-Bit I/O Expander to I2C ic module to interface with LCD display and microcontroller via I2C protocol. We selected address through A0,A1 and A2 pins.

PCF8574A

This is the internal structure of the PCF8574A IC.

LCD Display Circuit Diagram

The variable resistor on the left corner is used to change the display's contrast. However, we will not see this in our simulation. If we do this project practically, I must use this resistor. I use two pullup resistors because of the unstable state of the signal.

LCD Display Code

This is the code which we are going to use for the LCD display.

Switches

There are 2 switches in the remote control. One for the siren and one to shift between auto mode and manual mode. We does not use pullup resistors but we manage to prevent debouncing by make the pin as pullup from the microcontroller.

Version Controlling

We use GitHub as our git platform. We have included the Proteus project, simul IDE project, microchip studio project into the version controlling, and We have added all documents in the project also. Each one commits to their own branch. Finally, we code reviews peer-to-peer and merge them to the dev branch from a pull request. At last we merge the dev branch to master branch. Right now latest project is in the master branch.