NEW HORIZON
COLLEGE OF ENGINEERING

Autonomous College Permanently Affiliated to VTU, Approved by AICTE & UGC
Accredited by NAAC with 'A' Grade.

CONTACT-LESS PC CONTROL

A MINI PROJECT REPORT

Submitted by

PUDI YASHWANTH

PRAMOD AITHAL

VYSHAK SATHISH SHETTY

VARUN GOWDA K V

In partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING
IN
ELECTRONICS AND COMMUNICATION

2020-21

NEW HORIZON
COLLEGE OF ENGINEERING
Autonomous College Permanently Affiliated to VTU, Approved by AICTE & UGC
Accredited by NAAC with 'A' Grade.

BENGALURU-560103

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

CERTIFICATE

Certified that the Mini project entitled "Contact-less PC Control" is carried out by **Mr. Pudi Yashwanth** bearing USN: **1NH18EC134**, **Mr. Pramod Aithal** bearing USN: **1NH18EC087**, **Mr. Varun Gowda K V** bearing USN: **1NH18EC119** and **Mr. Vyshak Sathish Shetty** bearing USN: **1NH18EC123**, bonafide students of NHCE, Bengaluru in partial fulfillment for the award of Bachelor of Engineering in Electronics and Communication of the Visweswaraya Technological University, Belagavi during the year 2020-21. It is certified that all corrections and suggestions indicated for Internal Assessment have been incorporated in the report deposited in the department library. The mini project report has been approved as it satisfies the academic requirements in respect of the mini project work prescribed for the said degree

_____	_____
Signature of the Guide	Signature of the HoD
Mr Santhosh Krishna B V	Dr. Sanjeev Sharma
Sr. Assistant Professor Department of ECE	Professor & HoD Department of ECE
NHCE, Bengaluru	NHCE, Bengaluru

External Viva

Name of the Examiners	Signature with Date
1. _____	1. _____
2. _____	2. _____

ACKNOWLEDGEMENT

The satisfaction that accompany the successful completion of any task would be, but impossible without the mention of the people who made it possible, whose constant guidance and encouragement helped us succeed.

We thank **Dr. Mohan Manghnani**, Chairman of **New Horizon Educational Institution**, for providing necessary infrastructure and creating good environment.

We also record here the constant encouragement and facilities extended to us by **Dr.Manjunatha**, Principal, NHCE and **Dr. Sanjeev Sharma**, Head of the department of Electronics and Communication Engineering. We extend sincere gratitude to them.

We sincerely acknowledge the encouragement, timely help and guidance to us by our beloved guide **Mr. Santhosh Krishna B V, Senior AP-ECE,** to complete the project within stipulated time successfully.

Finally, a note of thanks to the teaching and non-teaching staff of Electronics and Communication Department for their cooperation extended to us, who helped us directly or indirectly in this successful completion of the mini project.

Pudi Yashwanth (1NH18EC134)

Pramod Aithal (1NH18EC087)

Vyshak Sathish Shetty (1NH18EC123)

Varun Gowda K V (1NH18EC119)

ABSTRACT

The world as we know today is taken over by a deadly virus known as COVID-19. As we have been the witness to this pandemic, we now know that a pandemic can suddenly appear out of nowhere and spread within the blink of an eye. There is no guarantee that this may not happen again. So it is our responsibility to be cautious and take as many precautions as possible.

We all use public computers for various reasons. It might be in internet cafes, to give presentations, in public libraries etc. Using public computers can be risky if we consider the factor of viruses as we never know the condition of the person who has used it before us. Thus we may be at a risk of catching the virus. It would be a lot more safer if we could just control the computers without really touching the computer. This can be done by making a few gestures at the computer. So we just have to wave a few signs in front of the computer and the task we wish to do will be completed without really touching the computer. This is what we aim to achieve with our project. Thus we call it the “Contactless PC Control”.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 01

INTRODUCTION

In this time of pandemic, we never know when or where we catch the COVID-19 virus. It is a microorganism, which means it is so small that it is practically invisible to the naked eye. The virus can be present on any surface and can stick to our hands when we touch those surfaces. If we touch our hands after that it can enter our body and multiply itself. COVID-19 is also highly infectious which means that it will spread very easily. So we should take precautions and avoid touching any surfaces when it is not necessary. We should take responsibility for our own as well as other's safety because when we have each other's back we are always safe.

Public places are where we are most vulnerable to the virus. Same applies for public computers. There are a lot of times that we have to use public computers but considering the COVID-19 scenario it is not really safe to use public computers. We never know who has used it before us and more importantly we do not know the status of the person who used it before. If he was not well, then there is a high chance that he may have left the virus on the places where he has touched the computer. If there was a way to control the computer without really touching it then it would be great for the prevention of spreading of the virus. Guess what? there is….

Our project device, the "Contactless PC Control", aims to let the user control the PC without actually touching it. Just by making a few gestures in the air we can control the PC. So the user does not need to use the keyboard or the mouse. If we do not touch the computer then the risk of us getting infected with the virus reduces dramatically. The device can be configured to perform different tasks based on different gestures. A combination of color can also be set if we wanted to. Moreover it is a portable device and can be connected to any computer we want. So we don't have to worry about a computer not having the device. We can carry the device

anywhere and plug it to any computer and it can immediately be controlled using gestures. This lets everyone carry their own gesture sensors and plug it in. When all of the users are not touching the computer and instead using gestures to control the computer, there is a very low chance (or even nil) of the virus spreading through computers.

This method of controlling the computer can also be the future of the computers. This technology may eliminate the use of keyboard and mouse completely. Everyone would be using computers with gestures only. Even the displays may be eliminated and holograms would be used instead. Gestures would compliment holograms very well as we would be able to control the holograms directly by touching it.

CHAPTER 02

LITERATURE SURVEY

Title of the paper & Journal	Author & Year of Publication	Outcome	Limitation
Basic understanding of the arduino.	M McRoberts 2011	Different pins in the arduino and their uses.	Gesture control using arduino is not explained in detail.
Hand Gesture Recognition.	Rafiqul Zaman Khan Noor Ibraheem 2012	General hand gesture recognition.	No connection with APDS 9960 module.
Touchless Hand Gesture Device Controller	Elliot Greenfield 2007	Controlling of different electrical devices using hand gestures.	Arduino is not used for gestures.

Real-Time Hand Gesture Recognition Using Finger Segmentation	Shan Zhao 2014	Method to recognize the hand gestures.	Uses finger segmentation for gesture control.
Hand Gesture Recognition Based on Computer Vision	Munir Oudah Ali AJavaan Chahl l-Naji	Using of the computer's camera to understand the gestures.	Uses computer vision for sensing gestures and not APDS 9960 module.

Table 2.1 Literature Survey

Inference :

Basic understanding of the gesture detection technology is realized. Also realized the basics of arduino and its different ports. Programming of the arduino can be done.

CHAPTER 03

PROPOSED METHODOLOGY

Existing Systems

The current system is of the traditional computer with a mouse/trackpad and keyboard being used to control the slides and pages during a presentation.as we know that both, the mouse and the keyboard require human contact to control them and cannot be controlled without having a physical contact, during this time of the pandemic,human contact is not advised as it can act as the source for the virus or the bacteria to spread.

While delivering a lecture or a speech the general trivia is to connect the system to the large screen and move back and forth while delivering the lecture and this looks really odd as it reduces the enthusiasm and interest in the audience.

Most of the laptops which are being released now do not come with the hdmi cable thus resulting in a big fuss as to how the presentation has to move forward. Therefore we have come up with a device that bypasses all these hardware obstacles and also makes the presentation easy and also attracts the crowd.

Problem Statement

The current system requires human contact to control the PC or device to accomplish tasks. That is we need to touch the keyboard and the mouse in order to perform tasks on the computer. The problem with this is the spread of some of the infectious viruses. This can be solved by controlling the computer without touching the computer which is the aim of this project.

Objectives:

- By using gestures we can minimize human contact.
- Chances of getting infected are less as there is no contact between the user and the device.
- Disabled or elderly people should be able to use this device without any hassle.
- Easy to understand and use.
- Using gestures to enhance the presentation and to keep the crowd engaged and involved.

Proposed methodology

The function of the device is pretty basic. The device uses the USB port to control the keyboard. Basic functionality of the computer can be done by using just the keyboard without a mouse. When a gesture is made on the sensor module it in turn clicks a button on the keyboard which is used to make certain functions.

When a person is using this, it looks much better while giving a presentation, the speaker will not have to walk back and forth to change slides, and look into the system to adjust the presentation, as this looks really odd while presenting to a mass crowd and it also bores the audience. When a speech is delivered, the speaker must attract the crowd and make them pay attention to whatever they say, thus resulting in a productive speech. To make this happen we planned on developing a device which controls the presentation through gestures.

In this device, we plan on using the APDS 9960 module to register the gestures that we will be performing.the APDS-9960 is a digital RGB, ambient light, proximity and gesture sensor device in a single 8-pin package.this can accurately sense simple UP-DOWN-RIGHT-LEFT gestures or more complex gestures. The addition of micro-optics lenses within the module provides high efficient transmission and reception of infrared energy. The gesture data will then be sent to an Arduino, the The Arduino we are using here is Arduino Integrated Development Environment(IDE). It takes the gestures that have been saved by the gesture module and using the pre written code here, it identifies the gesture and sends a signal to the presenting laptop. The laptop does the function and gives the desired output.

We plan on giving the basic operations to this system as to swipe left or right, maximize or minimize the screen and open and close files, as these are the basic operations while navigating the desktop. Other functionalities include changing slides in a powerpoint presentation, entering and exiting slideshow mode. This functionality is important because many people use public computers to give presentations. More functions can be added very easily just by tweaking the code a little bit. This makes the device portable as well as highly customizable.

Project Description

The current system is a traditional system which we generally use is a computer or laptop with a mouse or a trackpad and keyboard which is being used to control it. As we know, both mouse and keyboard require human contact to be operated and cannot be controlled without touching it, and during this time of pandemic, human contact is not really a good option as the virus can spread easily as the hardware components act as a carrier for the virus or the bacteria.

We have developed a device which requires minimum human contact to control the system as we will be using the gestures to accomplish the tasks on the computer or laptop.

In our project, we have coded our gestures in C language into the arduino using an ethernet cable as the coding is done on the laptop. We are dumping the code sequence from laptop to the arduino.

The sensor we are using to capture the gestures is APDS 9960, this sensor device features advanced Gesture detection, Proximity detection, Digital Ambient Light Sense (ALS) and Colour Sense (RGBC).

We have written the code for eight gestures that are commonly used while navigating through the computer for example to open and close files, to swipe left or right through the slides of the presentation, to scroll up and down on pages etc. We are using the proximity and distance to differentiate between the gestures.

Hardware Description

The project has a hardware and a software part. The hardware part is used to read the gestures that are performed. The software that we feed then interprets the gesture and performs the necessary function that is required. Below is the detailed hardware and software specifications.

Hardware Specifications

1. **APDS 9960**

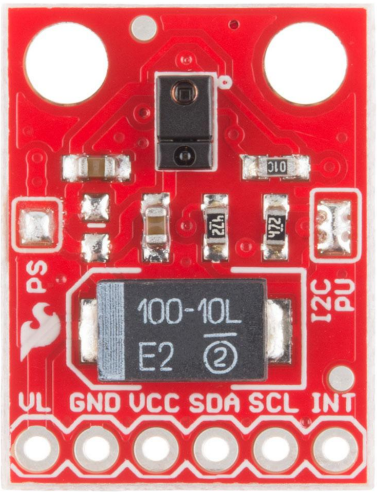


Fig 3.1 - Gesture sensor

The APDS-9960 (Fig 3.1) device features advanced Gesture detection, Proximity detection, Digital Ambient Light Sense (ALS) and Color Sense (RGBC). The slim modular package, L 3.94 x W 2.36 x H 1.35 mm, incorporates an IR LED and factory calibrated LED driver for drop-in compatibility with existing footprints.

Gesture detection

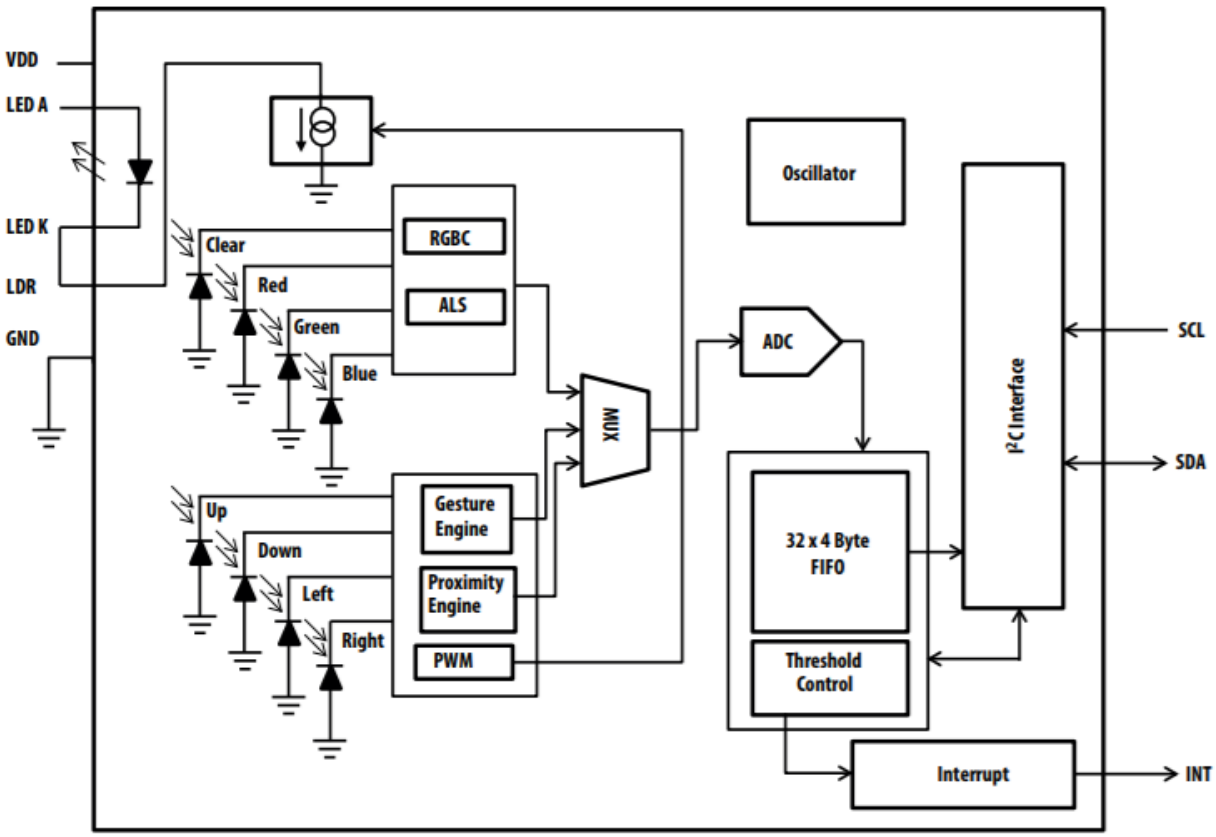
Gesture detection utilizes four directional photodiodes to sense reflected IR energy (sourced by the integrated LED) to convert physical motion information (i.e. velocity, direction and distance) to a digital information. The architecture of the gesture engine features automatic activation (based on Proximity engine results), ambient light subtraction, cross-talk cancelation, dual 8-bit data converters, power saving inter-conversion delay, 32-dataset FIFO, and interrupt driven I2C communication. The gesture engine accommodates a wide range of mobile device gesturing requirements: simple UP-DOWN-RIGHT-LEFT gestures or more complex gestures can be accurately sensed. Power consumption and noise are minimized with adjustable IR LED timing.

Proximity detection

The Proximity detection feature provides distance measurement (E.g. mobile device screen to user's ear) by photodiode detection of reflected IR energy (sourced by the integrated LED). Detect/release events are interrupt driven, and occur whenever proximity result crosses upper and/ or lower threshold settings. The proximity engine features offset adjustment registers to compensate for system offset caused by unwanted IR energy reflections appearing at the sensor. The IR LED intensity is factory trimmed to eliminate the need for end-equipment calibration due to component variations. Proximity results are further improved by automatic ambient light subtraction.

Color and ALS detection

The Color and ALS detection feature provides red, green, blue and clear light intensity data. Each of the R, G, B, C channels have a UV and IR blocking filter and a dedicated data converter producing16-bit data simultaneously. This architecture allows applications to accurately measure ambient light and sense color which enables devices to calculate color temperature and control display backlight.



Functional Block Diagram

Fig 3.2 - Block diagram

I/O Pins Configuration

Pin	Name	Type	Description
1	SDA	I/O	I²C serial data I/O terminal - serial data I/O for I²C-bus
2	INT	O	Interrupt - open drain (active low)
3	LDR		LED driver input for proximity IR LED, constant current source LED driver
4	LEDK		LED Cathode, connect to LDR pin when using internal LED driver circuit
5	LEDA		LED Anode, connect to V_{LEDA} on PCB
6	GND		Power supply ground. All voltages are referenced to GND
7	SCL	I	I²C serial clock input terminal - clock signal for I²C serial data
8	V_{DD}		Power supply voltage

Absolute Maximum Ratings over operating free-air temperature range (unless otherwise noted)*

Parameter	Symbol	Min	Max	Units	Conditions
Power supply voltage [1]	V_{DD}		3.8	V	
Input voltage range	V_{IN}	-0.5	3.8	V	
Output voltage range	V_{OUT}	-0.3	3.8	V	
Storage temperature range	T_{stg}	-40	85	°C	

Recommended Operating Conditions

Parameter	Symbol	Min	Typ	Max	Units
Operating ambient temperature	T_A	-30		85	°C
Power supply voltage	V_{DD}	2.4	3.0	3.6	V
Supply voltage accuracy, V_{DD} total error including transients		-3		+3	%
LED supply voltage	V_{LEDA}	3.0		4.5	V

Operating Characteristics, VDD = 3 V, TA = 25 °C (unless otherwise noted)

Parameter	Symbol	Min	Typ	Max	Units	Test Conditions
IDD supply current [1]	I_{DD}		200	250	µA	Active ALS state PON = AEN = 1, PEN = 0
			790			Proximity, LDR pulse ON, PPulse = 8 (I_{LDR} not included)
			790			Gesture, LDR pulse ON, GPulse = 8 (I_{LDR} not included)
			38			Wait state PON = 1, AEN = PEN = 0
			1.0	10.0		Sleep state [2]
V_{OL} INT, SDA output low voltage	V_{OL}	0		0.4	V	3 mA sink current
I_{LEAK} leakage current, SDA, SCL, INT pins	I_{LEAK}	–5		5	µA	
I_{LEAK} leakage current, LDR P\pin	I_{LEAK}	–10		10	µA	
SCL, SDA input high voltage, V_{IH}	V_{IH}	1.26		V_{DD}	V	
SCL, SDA input low voltage, V_{IL}	V_{IL}			0.54	V	

Proximity Characteristics, VDD = 3 V, TA = 25 °C, PEN = 1 (unless otherwise noted)

Parameter	Min	Typ	Max	Units	Test Conditions
ADC conversion time step size		696.6		µs	
ADC number of integration steps		1		steps	
Full scale ADC counts			255	counts	
LED pulse count [1]	1		64	pulses	
LED pulse width – LED on time [2]		4		µs	PPLEN = 0
		8			PPLEN = 1
		16			PPLEN = 2
		32			PPLEN = 3
LED drive current [3]		100		mA	LDRIVE = 0
		50			LDRIVE = 1
		25			LDRIVE = 2
		12.5			LDRIVE = 3
LED boost [3]		100		%	LED_BOOST = 0
		150			LED_BOOST = 1
		200			LED_BOOST = 2
		300			LED_BOOST = 3
Proximity ADC count value, no object [4]		10	25	counts	V_{LEDA} = 3 V, LDRIVE = 100 mA, PPULSE = 8, PGAIN = 4x, PPLEN = 8 µs, LED_BOOST = 100%, open view (no glass) and no reflective object above the module.

Gesture Characteristics, VDD = 3 V, TA = 25 °C, GEN = 1 (unless otherwise noted)

Parameter	Min	Typ	Max	Units	Test Conditions
ADC conversion time step size [1]		1.39		ms	
LED pulse count [2]	1		64	pulses	
LED pulse width – LED on time [3]		4		µs	GPLEN = 0
		8			GPLEN = 1
		12			GPLEN = 2
		16			GPLEN = 3
LED drive current [4]		100		mA	GLDRIVE = 0
		50			GLDRIVE = 1
		25			GLDRIVE = 2
		12.5			GLDRIVE = 3
LED boost [4]		100		%	LED_BOOST = 0
		150			LED_BOOST = 1
		200			LED_BOOST = 2 [5]
		300			LED_BOOST = 3 [5]
Gesture ADC count value, no object [6]		10	25	counts	V_{LEDA} = 3 V, GLDRIVE = 100 mA, GPULSE = 8, GGAIN = 4x, GPLEN = 8 µs, LED_BOOST = 100%, open view (no glass) and no reflective object above the module, sum of UP & DOWN photodiodes.
Gesture ADC count value [7, 8]	96	120	144	counts	Reflecting object – 73 mm × 83 mm Kodak 90% grey card, 100 mm distance, V_{LEDA} = 3 V, GLDRIVE = 100 mA, GPULSE = 8, GGAIN = 4x, GPLEN = 8 µs, LED_BOOST = 100%, open view (no glass) above the module, sum of UP & DOWN photodiodes.
Gesture wait step size		2.78		ms	GTIME = 0x01

Gesture Operation:

The Gesture detection feature [6] provides motion detection by utilizing directionally sensitive photodiodes to sense reflected IR energy sourced by the integrated LED. The following registers and control bits govern gesture operation and the operational flow is depicted in Figure

Register/Bit	Address	Description
ENABLE<PON>	0x80<0>	Power ON
ENABLE<GEN>	0x80<6>	Gesture Enable
GPENTH	0xA0	Gesture Proximity Entry Threshold
GEXTH	0xA1	Gesture Exit Threshold
GCONFIG1<GFIFOTH>	0xA2<7:6>	Gesture FIFO Threshold
GCONFIG1<GEXMSK>	0xA2<5:2>	Gesture Exit Mask
GCONFIG1<GEXPERS>	0xA2<1:0>	Gesture Exit Persistence
GCONFIG2<GGAIN>	0xA3<6:5>	Gesture Gain Control
GCONFIG2<GLDRIVE>	0xA3<4:3>	Gesture LED Drive Strength
GCONFIG2<GWTIME>	0xA3<2:0>	Gesture Wait Time
STATUS<PGSAT>	0x93<6>	Gesture Saturation
CONFIG2<LEDBOOST>	0x90<5:4>	Gesture/Proximity LED Boost
GOFFSET_U	0xA4	Gesture Offset, UP
GOFFSET_D	0xA5	Gesture Offset, DOWN
GOFFSET_L	0xA7	Gesture Offset, LEFT
GOFFSET_R	0xA9	Gesture Offset, RIGHT
GPULSE<GPULSE>	0xA6<5:0>	Pulse Count
GPULSE<GPLEN>	0xA6<7:6>	Gesture Pulse Length
GCONFIG3<GDIMS>	0xAA<1:0>	Gesture Dimension Select
GCONFIG4<GFIFO_CLR>	0xAB<2>	Gesture FIFO Clear
GCONFIG4<GIEN>	0xAB<1>	Gesture Interrupt Enable
GCONFIG4<GMODE>	0xAB<0>	Gesture Mode
GFLVL	0xAE	Gesture FIFO Level
GSTATUS<GFOV>	0xAF<1>	Gesture FIFO Overflow
GSTATUS<GVALID>	0xAF<0>	Gesture Valid
GFIFO_U	0xFC	Gesture FIFO Data, UP
GFIFO_D	0xFD	Gesture FIFO Data, DOWN
GFIFO_L	0xFE	Gesture FIFO Data, LEFT
GFIFO_R	0xFF	Gesture FIFO Data, RIGHT
CONFIG1<LOWPOW>	0x8D	Low Power Clock Mode

2. Arduino

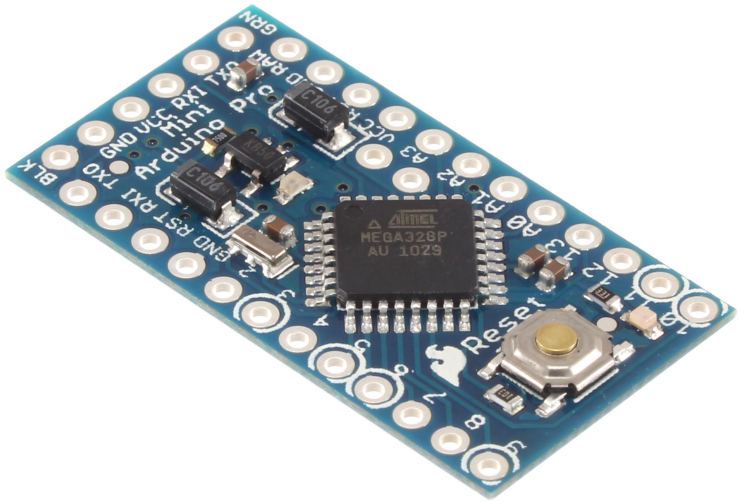


Fig 3.3 - Arduino Pro Micro

We use the Arduino (Fig 3.3) for processing the data that is received by the APDS 9960 module. The code is also feeded into the arduino. So the arduino takes the gesture data and then uses the code to execute the commands that the user wishes to perform.

In the “Contactless PC Control” we use the Arduino Pro Mini. This board is also manufactured by SparkFun, the same manufacturer as the APDA 9960 module.

This board was developed for applications and installations where space is premium and projects are made as permanent set ups. Small, available in 3.3 V and 5 V versions, powered by ATmega328

specs:

Microcontroller	ATmega328 *
Board Power Supply	3.35 -12 V (3.3V model) or 5 - 12 V (5V model)
Circuit Operating Voltage	3.3V or 5V (depending on model)
Digital I/O Pins	14
PWM Pins	6
UART	1
SPI	1
I2C	1
Analog Input Pins	6
External Interrupts	2
DC Current per I/O Pin	40 mA
Flash Memory	32KB of which 2 KB used by bootloader *
SRAM	2 KB *
EEPROM	1 KB *
Clock Speed	8 MHz (3.3V versions) or 16 MHz (5V versions)

Power

The Arduino Pro Mini can be powered with an FTDI cable or breakout board connected to its six pin header, or with a regulated 3.3V or 5V supply (depending on the model) on the Vcc pin. There is a voltage regulator on board so it can accept voltage up to 12VDC. If you're supplying unregulated power to the board, be sure to connect to the "RAW" pin on not VCC.

The power pins are as follows:

RAW For supplying a raw voltage to the board.

VCC The regulated 3.3 or 5 volt supply.

GND Ground pins.

Memory

The ATmega328 has 32 kB of flash memory for storing code (of which 0.5kB is used for the bootloader). It has 2 kB of SRAM and 1kBs of EEPROM (which can be read and written with the EEPROM library.

Input and Output

Each of the 14 digital pins on the Pro Mini can be used as an input or output, using pinMode,digitalWrite, and digitalRead functions. They operate at 3.3 or 5 volts (depending on the model). Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

Serial: 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the TX-0 and RX-1 pins of the six pin header.

External Interrupts: 2 and 3. These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the attachInterrupt function for details.

PWM: 3, 5, 6, 9, 10, and 11. Provide 8-bit PWM output with the analogWrite function.

SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins support SPI communication, which, although provided by the underlying hardware, is not currently included in the Arduino language.

LED: 13. There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off

The Pro Mini has 8 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). Four of them are on the headers on the edge of the board; two (inputs 4 and 5) on holes in the interior of the board. The analog inputs measure from ground to VCC. Additionally, some pins have specialized functionality:

I2C: A4 (SDA) and A5 (SCL). Support I2C (TWI) communication using the Wire library.

There is another pin on the board:

Reset. Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

Communication

The Arduino Pro Mini has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega328 provides UART TTL serial communication, which is available on digital pins 0 (RX) and 1 (TX). The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board via a USB connection.

A SoftwareSerial library allows for serial communication on any of the Pro Mini's digital pins.

The ATmega328 also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; see the reference for details. To use the SPI communication, please see the ATmega328 datasheet.

Programming

The Arduino Pro Mini can be programmed with the Arduino software download. For details, see the reference and tutorials.

The ATmega328 on the Arduino Pro Mini comes pre burned with a bootloader that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol reference , C header files.

You can also bypass the bootloader and program the ATmega328 with an external programmer; see these instructions for details.

Automatic (Software) Reset

Rather then requiring a physical press of the reset button before an upload, the Arduino Pro Mini is designed in a way that allows it to be reset by software running on a connected computer. One of the pins on the six-pin header is connected to the reset line of the ATmega328 via a 100 nF capacitor. This pin connects to one of the hardware flow control lines of the USB-to-serial convertor connected to the header: RTS when using an FTDI cable, DTR when using the Sparkfun breakout board. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of the reset line can be well-coordinated with the start of the upload.

This setup has other implications. When the Pro Mini is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Pro. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.

Software Description

Arduino IDE

Fig 3.4 - Code Editor

The Arduino IDE (Fig 3.4) is an open-source software using which we can write code and easily push it to the Arduino board. This IDE is cross platform meaning that it will work on Windows, Mac OS as well as Linux.

We use the program to write the code. We can also use C++. the program when executed takes the gestures from the APDS9960 module and uses the data given by the module to execute the commands according to the code written.

Code:

```
#include <Arduino_APDS9960.h>
#include "Keyboard.h"

void setup() {
  Serial.begin(9600);
  if (!APDS.begin()) {
    Serial.println("Error initializing APDS9960 sensor!");
       }
   Keyboard.begin();
  Serial.println("Detecting gestures ...");
}

void loop() {
  if (APDS.gestureAvailable() && APDS.proximityAvailable())
   {
    int gesture = APDS.readGesture();
    int prox = APDS.readProximity();
    int set;
    if(prox>=0 && prox<=70) set=1;
    if(prox>=100 && prox<=255) set=2;

  switch(set) {
        case 1:
            switch(gesture)
                case GESTURE_UP:
                Keyboard.write(KEY_UP_ARROW);
                break;

                case GESTURE_DOWN:
                Keyboard.press(KEY_DOWN_ARROW);
                break;

                case GESTURE_LEFT:
                Keyboard.write(KEY_LEFT_ARROW);
                break;

                case GESTURE_RIGHT:
                Keyboard.write(KEY_RIGHT_ARROW);
                break;

                default: break;
             break;

        case 2:
             switch(gesture)
```

```
                case GESTURE_UP:
                Keyboard.write(KEY_RETURN);
                break;

                case GESTURE_DOWN:
                Keyboard.press(KEY_LEFT_ALT);
                Keyboard.write(KEY_F4);
                Keyboard.release(KEY_LEFT_ALT);
                break;

                case GESTURE_LEFT:
                Keyboard.write(KEY_F5);
                break;

                case GESTURE_RIGHT:
                Keyboard.write(KEY_ESC);
                break;

                default: break;
            break;
    }
  }
}
```

Code Brief:

The code splits the space above the sensor into parts, set 1 and set 2 . set 1 is any movement near to the sensor and set 2 is any movement away from the sensor. So what the code does is gets two variables from the sensor. Distance of the hand from the sensor and the gesture movement. Now combining these two variables we can create a total of 8 cases. 4 gestures near the sensor and 4 gestures away from the sensor. Then we assign different key strokes to the different cases available. Therefore we can see that at the current configuration level we can perform 8 different key functions with the gestures which is enough to do some basic navigation and a powerpoint slideshow presentation. However we can always add more functionalities by changing the code. Instead of splitting the space into two different parts, we can split it into three parts, hence giving us 12 different keystrokes possible. If we split it into 4 parts we get 16 different functions possible. So if we consider the number of splits as N then we can perform 4N functions. So according to the users requirements we can tune the program to

fulfill the needs of the user that is required to successfully control the computer without touching it.

Working of the system

The APDS9960 module is a gesture sensing module and gets the input from our hand movement over the sensor. There are four types of movements that it can sense, up, down, right and left. Apart from this the module can also see the proximity of the object from the sensor as well as RGB colour. The APDS9960 module sends all the data collected to an Arduino Pro Mini. The arduino contains the code which decides which function to perform based on what gesture. So the arduino is used to process the data that is collected by the APDS9960 module.

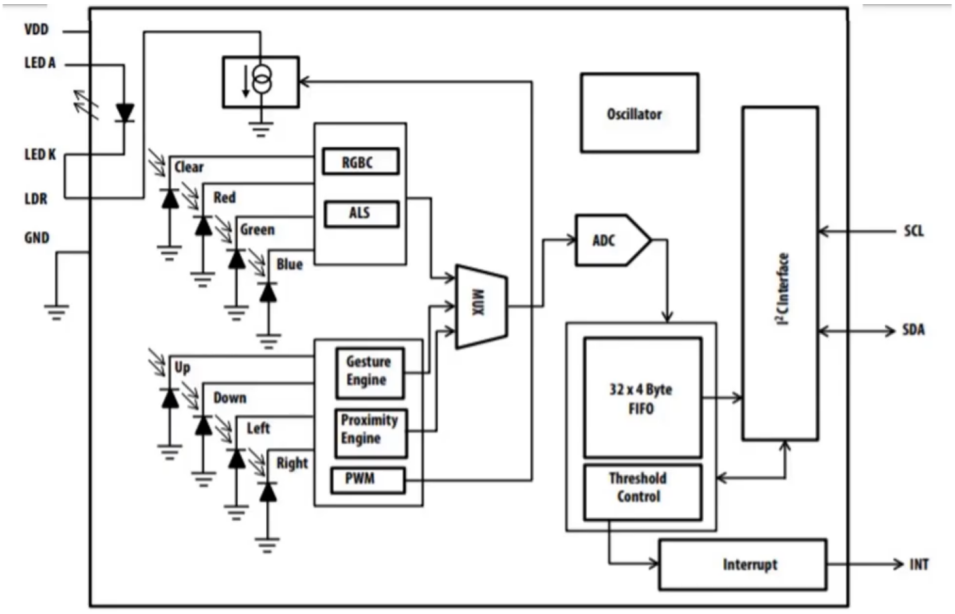


Fig 3.5 - Block diagram of apds9960

This module has multiple diodes which is used to sense the proximity and the direction of movement. Using the combination of the diodes, the sensor can detect the direction of the movement and the proximity of the object. This sensor has two separate sets of diodes to check colour and direction respectively. The colour diode set has 4 diodes. They collect red, green and blue colours. The third diode is used for no color detection. That is clear space on top of the sensor. Similarly in the set of diodes that is used to detect direction, there are 4 sensors. They

sense the directions left, right, up and down. The proximity sensing functionality does not use a separate set of diodes but uses the existing set itself.

Once the direction, colour and proximity is detected, all that information is then feeded into the arduino for processing. The arduino will be powered by the laptop that it is connected to. It will contain the code which we use to program the arduino. The code utilizes the data that is provided by the APDS9960 module. The arduino that is also feeded with this code is the one that interprets the data and utilizes the data according to the code written. The code can be customised as per our requirement thus expanding the functionality of this device. As the changes in the code is simple and quick to update in the arduino, it can be done almost immediately. After the arduino processes all the information, it now uses this information to inturn control the keyboard of the computer. Using basic keys as well as the various keyboard shortcuts available to us, we can practically do anything in the computer without using the mouse. Now since we can control the keyboard with our gestures through the APDS 9960 module and the Arduino, we are able to control the computer with different gestures. And by combining the colour, proximity and direction of movement, we can widen the amount of functionality. So we are not just confined to the four gestures and four keys but there are infinite possibilities by using and combining three different factors together.

CHAPTER 04

RESULTS AND DISCUSSION

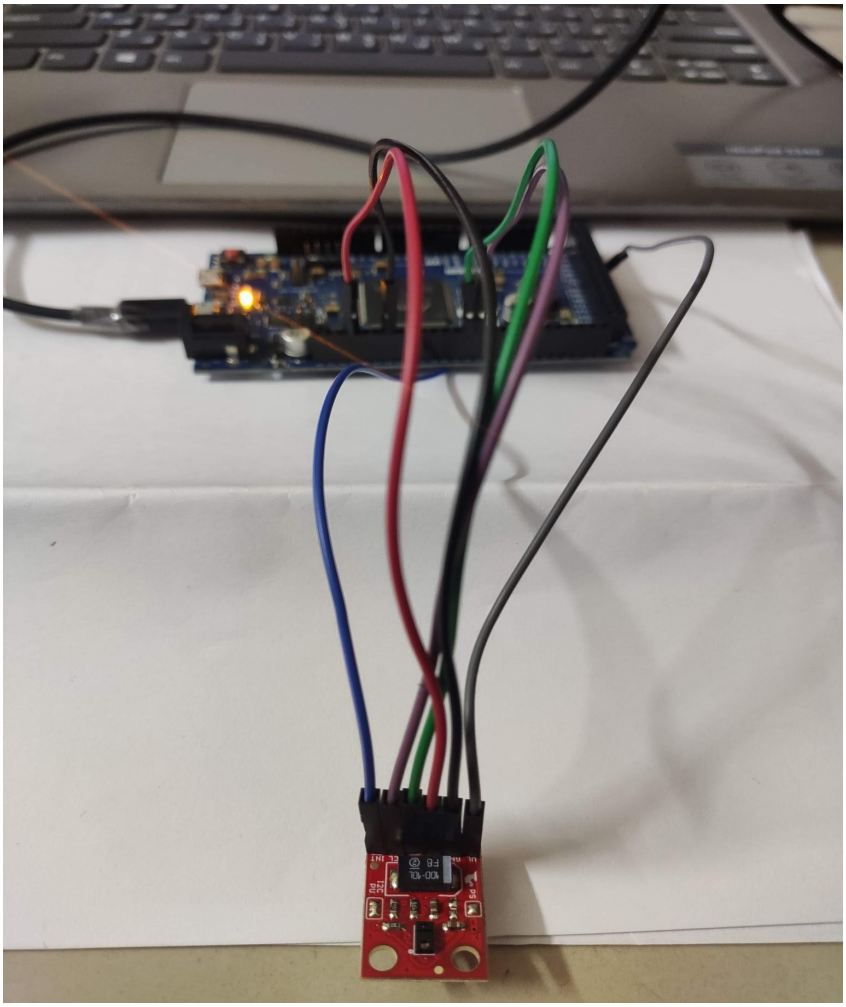


fig 4.1 - Model Image

The “Contact-less PC Control” is a good alternative to the traditional mouse and keyboard setup when the factor of the COVID-19 pandemic is taken into picture. The pandemic has taken our society by storm and it has spread like wildfire. We never know on what surface the virus can lie. So it is necessary to taken the proper precautions. We should avoid touching commonly used surfaces like public computers. So this device can be used in place of a keyboard and a mouse to control the computer. This device can be used to control the computer without touching it by

using certain gestures to control the computer. Hence we can avoid touching the computer and prevent the spread of the virus.

This device is also compact and portable so it can be carried around easily. It can also be connected to any computer and immediately that computer can be controlled by using gestures without any changes in the code.

Advantages of the project:

- By using gestures we can minimize human contact.
- Chances of getting infected are less as there is no contact between the user and the device.
- Disabled or elderly people should be able to use this device without any hassle.
- Easy to understand and use.
- Using gestures to enhance the presentation and to keep the crowd engaged and involved.
- It can be plugged to any PC and we would be able to control that PC using gestures.
- The response time is fast.
- It is a low cost operating device.
- More accurate.
- High stability.

Applications of the project:

- While delivering speeches or presentations.
- Can be used in defence.
- It can be used in home automation.
- To control the belt flow in manufacturing goods.
- It can also be used in controlling the traffic lights by the officer instead of touching the switches everytime.

Chapter 05

CONCLUSION AND FUTURE SCOPE

Conclusion:

We have created a device in which we can reduce the fear of germs called germophobia and also the spreading of the virus during this time of pandemic.

Using this technology we can make progress towards a futuristic world where the computer peripherals are just our hands and we interact with the software like we would with physical objects.

Future scope:

The components can be made portable so that it can be easily carried around to be connected to a public PC. The contact of germs or viruses in public computers like libraries can be eliminated if everyone uses this wireless gesture device to control.

In future the device can be made smaller and portable and can also be made faster to detect gestures.

REFERENCES

Links:
[1]https://cdn.sparkfun.com/assets/learn_tutorials/3/2/1/Avago-APDS-9960-datasheet.pdf (opened on 18/01/2021)

[2] www.arduino.cc (opened on 18/01/2021)

[3] https://www.arduino.cc/en/Reference/ArduinoAPDS9960 (opened on 18/01/2021)

[4]https://www.arduino.cc/reference/en/language/functions/usb/keyboard/keyboardmodifiers/ (opened on 18/01/2021)

Books:
[4] Basic Understanding Of The Arduino, M McRoberts, 2011

Papers:
[5] Rafiqul Zaman Khan, Noor Ibraheem, "Hand Gesture Recognition: A Literature Review", in International Journal of Artificial Intelligence & Applications (IJAIA) Vol 3, 2012/08/01

[6] Elliot Greenfield, "Touchless hand gesture device controller", in Publication of US20080256494A1, 2007

[7] Zhi-hua Chen, Jung-Tae Kim, Jianning Liang, Jing Zhang, Yu-Bo Yuan, "Real-Time Hand Gesture Recognition Using Finger Segmentation", The Scientific World Journal, vol. 2014, Article ID 267872, 9 pages, 2014. https://doi.org/10.1155/2014/267872

[8] Oudah, Munir; Al-Naji, Ali; Chahl, Javaan. 2020. "Hand Gesture Recognition Based on Computer Vision: A Review of Techniques" J. Imaging 6, no. 8: 73.