CHALMERS

KeyTrack

Combining the keyboard with a trackpad

Joakim Bergström, Fredrik Ehrndal,
Filip Hesslund, Piérre Reimertz, David Spånslätt

Departments of Computer Science & Engineering and Information Technology
Chalmers University of Technology
Gothenburg, Sweden, June 2013
Bachelor's Thesis 2013:34

KeyTrack
Combining the keyboard with a trackpad

Joakim Bergström,
Fredrik Ehrndal,
Filip Hesslund,
Piérre Reimertz,
David Spånslätt

Examiner: Arne Linde

Chalmers University of Technology
University of Gothenburg
Department of Computer Science and Engineering
SE-412 96 Göteborg
Sweden
Telephone + 46 (0)31-772 1000

Department of Computer Science and Engineering
Göteborg, Sweden June 2013

Abstract

In today's computer-centered world, there is a lot of problem with repetitive strain injuries. This thesis describes a project aimed to develop a new, more ergonomic approach to control the computer, called KeyTrack.

The thesis describes the development and implementation of a prototype following certain requirements concerning i.e. accuracy and functionality. The approach taken to solve the problem is using a field of infrared light to find movements made within this field.

Several prototype iterations were built, each of them were improvements of the former. The final prototype developed, showed that there are certain problems with the infrared light approach, namely "light-noise" coming from other sources than the dedicated emitters.

The prototype is functional although it requires improvements to be able to replace the mouse and keyboard or a trackpad.

Sammandrag

I dagens värld har persondatorer en central roll, det finns mycket problem med exempelvis musarm. Denna rapport beskriver ett projekt som riktat in sig på att utveckla ett nytt, mer ergonomiskt sätt att styra datorn, kallat KeyTrack.

Rapporten beskriver utvecklingen och implementationen av en prototyp som utvecklats med krav angående användarvänlighet och funktionalitet. Ansatsen som togs för att lösa problemet var att med hjälp av ett fält av infrarött ljus registera rörelser i fältet.

Ett flertal iterationer av prototyper byggdes, där varje prototyp var en förbättring av den föregående. Den slutgiltiga prototypen visade på att det finns vissa problem med att använda infrarött ljus, nämligen störningar som kommer från oönskade ljuskällor.

Prototypen fungerar men behöver förbättras för att kunna ersätta tangentbord och mus eller musplattan.

Acknowledgements

The project group would like to thank Lars Bengtsson and Arne Linde for their coaching during the length of the thesis work, and also Morten Fjeld for his valuable input.

The Authors, Gothenburg; June 6, 2013

List of Abbreviations

RSI repetitve strain injury

IR Infrared

CCR Container code regognition

USB Universal serial bus

LED Light emitting diode

LCD Liquid crystal display

TFT Thin-film transistor

GUI Graphical user interface

Contents

List of Figures

1

Introduction

The first computer mouse was invented by *Tom Cranston*, *Fred Longstaff* and *Kenyon Taylor* as a secret military project in 1952.[1] Public introduction would have to wait until 1968 when Douglas Engelbart held the now famous *mother of all demos*[2], which introduced a majority of what today is considered as basic computer interaction design and more importantly, the computer mouse.

Since then, not much has changed to the fundamental design of the computer mouse, which would not be a problem if the computers of today would be used as Engelbart predicted. He predicted that each hand would be placed in a fixed position, one hand on a specialized keyboard and the other hand on the computer mouse.[3]
However, this is usually not the way one interacts with a computer. The user either types on the keyboard using both hands, or uses one hand to control the mouse pointer. These movement patterns could lead to repetitive stress injuries.[4]

1.1 Purpose and aim

The purpose of this bachelor project is to design and develop a more ergonomic[5] and space efficient alternative to the regular computer mouse. This by developing a prototype that centralizes hand movement and eliminates unnecessary repetitive movements.

All parts and materials used should be at as low cost as possible. This is to secure further development after the bachelor project time has ended, to make sure if possible, that future manufacturing of a product could be applicable.

1.2 Delimitations

This project was only focused on the ergonomic usage of computer peripherals, and not the other aspects of ergonomically improving the workstation. The project did not focus on integrating the solution fully into a keyboard. Rather, it is an extra peripheral placed around the keyboard.

1.3 Method

The project was divided into three main phases, a literature study, an implementation phase and a test phase. The phases gave the project structure and made it easier to have a general idea of the time schedule. During the literature phase information was gathered. The implementation phase was when the development took place along with most of the report writing. The last phase was a test phase when the prototype was tested and the results analyzed along with finalizing the report.

2

Idea

During prolonged use of either a desktop-computer or laptop, a user is likely to get minor aching pain in wrists and/or arms etc.[5] One reason for this could be that the keyboard usually is separated from the pointing device.

The idea of this project is to develop a prototype that solve ergonomic problems related to human-computer interaction without affecting the keyboard-layout and design. The prototype should be a combination between a keyboard and the mouse.

2.1 Design

The design is meant to solve the ergonomic issues of a standard desktop setup . When looking at what brings the most common work-related injuries, it becomes clear that an integration between mouse and keyboard could be beneficial.[5]

When exploring ways to integrate the keyboard and mouse, the idea of having a frame with some kind of motion sensors came to mind. The use of motion sensors was inspired by the LeapMotion device[6], but as a more keyboard-focused approach. This is why a frame around the keyboard was conspicuous.

When putting a frame on top of a keyboard it is important to not make the hands suffer from it, therefore the frame should be adjusted on the bottom side which will keep your hands and wrists comfortable. It is clear that the design needs to be as smooth as possible which has to be considered when constructing the prototype.

The best design choice from an ergonomic aspect would be to integrate it completely with a keyboard, the way a trackpad is integrated with a laptop. This would be an ever more comfortable design than with a small frame, since the only peripheral required is a

Figure 2.1: A possible design of the prototype

keyboard. However, this approach would be more complex than what could be achieved within the project's time limit, since it would involve rebuilding a laptop or a keyboard.

Figure 2.2: A future design of the product.

The idea of the project included a way of shutting down areas along the frame surface. This could be done using a graphical interface. An example would be if the user preferred to only use ones right hand for mouse movements, one could with ease shut down the left part of the frame to avoid any complications.

2.2 Interaction/movement patterns

When interacting with the computer, the mouse cursor would be controlled by finger movements. If the movements are natural enough there is not much need for learning any new techniques to move the cursor. The finger movements should not be registered when typing on the keyboard to avoid any complications and increase the workflow.

3

Theory

This chapter presents the ergonomic issues that comes with desktop and laptop computers. Several suggestions on how to type properly to increase your workflow is included. Lastly, products that address and tries to solve the ergonomic problems as well as the projects considered approaches are found in this chapter.

3.1 Ergonomics

The computer has become an important tool both in leisure and in the workplace. For a lot of people, this means that several hours a day are spent in front of a computer. To be able to do this, it is crucial that the workstation and how the user interacts with the computer does not lead to injuries. There are many guidelines and instructional videos on how to work with a computer in an ergonomic way, but it is difficult to always remain in an healthy position during a prolonged session in front of the computer. This opens up for alternatives to the classic computer mouse that helps the user to work in an ergonomic way.

3.1.1 Repetitive strain injuries

Repetitive strain injuries, also known as RSI, are "injuries to the musculoskeletal and nervous systems that may be caused by repetitive tasks ... ".[4] One could think that RSI is a modern phenomenon, but they have been recorded in medical literature for a long time. It was first described in 1700 by the Italian physician *Bernardino Ramazzini*.[7]

Using a classic computer mouse can result in injuries since the body, every time you want to move the mouse needs to reposition. Each time this is done, it causes the back to lean forward, the arm to extend and usually this arm is also used to support the weight of the body.

When moving the mouse, the wrist and hands may be strained by the small repetitive movements necessary. By minimizing these movements or possibly even eliminating them, the risks for RSI will be decreased[8].

3.1.2 Ways to minimize the risks

In a work environment it is important to inform about injuries that can occur. If you are unaware of the physical injuries, you are likely to get them eventually. There are simple ways to prevent these injuries from happening, the information presented below comes from a site dedicated to RSI related issues:[9]

- Keep a good posture when seated. The head and back should form a straight line from the ears to the pelvis.
- Avoid keeping the wrist bent to one side when typing, try to keep them in a straight line with the forearm.
- Don't hit the keys on the keyboard too forcefully, keep the typing relaxed
- Try to use all the fingers while typing and try to avoid looking at the keyboard. This will lessen the load on specific fingers and keeps your focus from the keyboard.
- Keep the grip around the mouse soft and place it close to the keyboard. This is to avoid reaching for it.
- Use a headset if you take a lot of phone calls. Avoid reaching for a phone.
- Use ergonomic equipment (for example the Microsoft ergonomic keyboard, Figure 3.1).
- Along with this it is recommended to use keyboard shortcuts, this will increase the effectiveness and decrease the need for a computer mouse which will minimize the amount of awkward body positions.

3.1.3 Keyboard usage guidelines

There are a numerous guidelines available for users and here, a recommendation from the computer company Logitech is presented.

Comfort

Regardless of how much time one spend using a keyboard, there are recommended ways to stay comfortable and productive. Physical discomfort, nerve injuries and muscle injuries may arise after long periods of repetitive movement using a poorly designed workspace, wrong body position or bad work habits.

When using a keyboard it is important to keep the shoulders, arms, wrist and hands relaxed and comfortable. This can be achieved by, for example, taking a break and letting the arms hang loosely at the sides for a moment. Avoid placing the hands or supporting the wrists on sharp edges on the desk. It is important to stop typing if pain, weakness, numbness or other similar symptoms arise.[10]

Effective typing

A keyboard consist of more than 100 keys so it is very important to position the hands in a correct way. The keyboard is divided into four different areas, the alphanumeric keypad or the "typing area", the function keys, the numeric keypad, and the navigation keypad.[10]

Figure 3.1: An ergonomic keyboard from Microsoft.[11]

The optimal hand position is to place the left index finger on the 'F' key and the right index finger on the 'J' key. From this position the left hand is naturally covering the left side of the keyboard and the right hand is covering the right side. The reason for this is to make all fingers useful when typing.[12] It's not uncommon to use the index fingers to press all the keys, but this is not optimal.

Logitech is a company that sells various computer related products. Amongst them is the standard computer mouse and keyboard. On their website they've published a guide on how to use keyboard and mouse properly to avoid injuries and to work more efficiently.[10]

3.2 Related work

This section gives a short introduction to alternative solutions on how to interact with a computer. The devices and solutions presented here are those interesting to this project and those discussed in the literature phase.

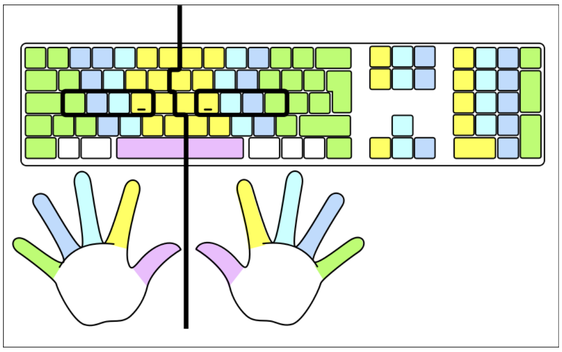


Figure 3.2: Fingers and parts of the keyboard; color-coded.[13]

There are numerous alternatives to the classic computer mouse. Most of these are designed with a focus on improving the user's work environment in an ergonomic way. The most common alternatives used today are the trackpad and the trackball mouse.

3.2.1 Trackpad

The trackpad can usually be found on laptops but can also be a standalone product. The on screen pointer is controlled by sliding the finger over a touch sensitive area and multiple gestures can be made to do different actions. The amount of movement needed is greatly decreased when comparing a trackpad with a classic computer mouse.[14]

Figure 3.3: The trackpad of an HP laptop.[15]

3.2.2 Trackball

The trackball mouse looks similar to the classic computer mouse, but instead of navigating the mouse by moving the whole mouse, navigation is made by rotating a small ball positioned on top of the mouse. This eliminates the need to move the arm and only small movements with one finger is needed to control the mouse cursor.[16]

Figure 3.4: A trackball from Logitech.[16]

Even though the trackball minimizes the hand movements when controlling the mouse, the positioning is still the same as with a standard computer mouse. This means that the user still needs to reach for the trackball.

3.2.3 LeapMotion

On the 22nd of July 2013 a company is releasing an alternative called LeapMotion. LeapMotion is a device which uses IR and CCR cameras to detect your hand movements in a 3D space. Due to protection of the product, there is limited information about the functionality.[17][6]

When looking at the LeapMotion's promotional video[19], little is said on how it could work alongside a keyboard. From an ergonomic aspect it is hard to see comfort in the suggested hand movements during prolonged use. It seems as though it is primarily meant to replace the mouse under certain conditions, and not during *all* kinds of work.

Figure 3.5: LeapMotion.[18]

3.2.4 RollerMouse

The RollerMouse has been on the market for about a decade and has become increasingly popular. The idea is to centralize body movements, minimize the usage of the shoulders and back, and use as few muscles as possible. The product is positioned in front of the keyboard and is used by moving a rollerbar with the fingers, thus moving the cursor. The RollerMouse has wrist support, keys for mouse clicks and a mouse scroll[20][21].

Figure 3.6: RollerMouse with higlighted points of interest.

The central part highlighted in red in Figure 3.6, is the rollerbar used to move the mouse cursor. This is controlled by the fingers of the user and is also clickable. As can be seen above, the keyboard is placed in the blue area on top of the RollerMouse, thus integrating it with the keyboard.

The benefits of the RollerMouse is interesting for the project, since the approach to the ergonomic problem was similar to the idea of KeyTrack; centralizing hand movements, hence involving less muscles for a more ergonomic computer session.

3.2.5 Tobii PCEye

Tobii PCEye is an alternative way of controlling a computer. It tracks eye movements to execute commands. It is primarily meant for people who are not physically able to control the PC with conventional methods like mouse and keyboard. Tobii PCEye features an internal processor and is connected to the PC with a USB-cable.[22]

Figure 3.7: Tobii PCEye.[23]

3.2.6 Motion sensing input device

An increasingly popular way of human interaction with computers is with motion capture, also referred to as performance capture or motion tracking. Motion capture is the process of recording objects, reference points and even humans, translating them to digital input. It has a broad use and is used in film-making, medical and military applications, with minor different approaches and implementations. Although it originates from the game development industries, the motion sensing could potentially act as an alternative to the regular computer interaction.[24].

The Nintendo Wii^{TM} game console uses a technique to capture motions made by the users hand for input to the console. It uses a game controller called the Wii Remote (or WiiMote) that uses infrared light signals to triangulate the location of the remote included with an angle [25]. The information is then used as input to the game console and translated into signals providing a location for a mouse cursor etc. Thus making it an alternative for digital interaction. The system uses a sensor bar in front of the capturing target to locate motions. It also contains an pivotal accelerometer that register movements on the x-axis, y-axis and z-axis. Enabling it to register movements up, down, in sideways and front to back, all in all the whole 3D spectrum for motions [26].

3.2.7 DGTS: Integrated Typing and Pointing

In a project done at Chalmers University of Technology and at ETH Zurich, a group implemented a keyboard with integrated mouse-controlling functionality[27]. This was done using a capacitive surface and a blob detection algorithm. In the promotional video, a very good accuracy and controllability is shown.

3.3 Considered approaches

A number of approaches were considered before deciding on the current solution (see page 15 for a more detailed insight on the current design and 32).

3.3.1 Laser

An idea considered was having a focused beam of light traversing the area above the keyboard back and forth at a high speed, using sensors and getting the position of an obstruction. This could be implemented using i.e. a laser pointer and a rotating prism to angle the beam to cover the whole keyboard. This seemed like an unnecessarily complex setup and therefore was discarded.

3.3.2 Ultrasound

Ultrasound transceivers could be used to measure the distance from a frame around the keyboard to a finger near the middle of the keyboard. One could then use trilateration to determine the obstructions position on a coordinate system.

The problem with ultrasound however was that the transceivers within projects budget did not work properly on distances less than about 30cm. This resulted in the dismissal of an ultrasound solution.

3.3.3 Glove approach

One could use a glove containing a gyro-accelerometer to measure how the hand moves, and in such a way controlling the computer by pointing at the screen and waving with the hand. One could complement this glove with using the webcamera that most laptops has at the top of the screen to locate the fingers positions.

This idea seemed good but the glove had to be tailored for different people. Both by size and by left-/right hand preference. It also seemed like a very expensive approach and was therefore discarded.

3.3.4 Infrared light

Infrared light emitters and receivers are relatively cheap compared to the approaches discussed above, and are easily placed in a circuit. There are many variants of receivers that react to different wavelengths of light. IR-LEDs have proved to be fail-proof, see more details in section 4.4.3).

4

Prototype

This chapter contains the realisation of the idea that was formed in the earlier chapters. Requirements are introduced to set standards on how the prototype should work. The design choices, a survey, hardware components and software design are stated in this chapter to clarify how the prototype is constructed in detail.

4.1 Requirements

It is crucial when developing an alternative to the computer mouse that it runs smoothly and that it can provide all the functionality that a classic computer mouse has. The focus of KeyTrack was to decrease the risk of injuries, however, to make sure it was a viable option, it had to be easy and comfortable to use. To achieve this, the development of KeyTrack was carried out with these requirements in mind.

4.1.1 Responsiveness

The KeyTrack needs to be responsive. When the user performs an action, it should instantly yield results on the monitor.

4.1.2 Accuracy

The KeyTrack needs to be accurate, even small movements needs to be registered and executed in a correct way. The user should be able to control the mouse in small sections without an issue.

4.1.3 Functionality

The KeyTrack needs to offer the same functionality as a classic computer mouse. This includes the basic movements, left and right clicks, being able to scroll up and down and select areas of the screen.

4.1.4 Customization

The usability of the KeyTrack can be increased by offering the user a possibility to customize certain aspects of how KeyTrack functions. To be able to, for example, set the sensitivity or choose which areas of the keyboard that can be used to control the mouse, it is important to increase the user experience.

4.2 Survey

A form-based survey was conducted in a group of 48 participants to confirm basic design choices such as enclosure and placement. The results of the survey gave important information about the target group. The target user is a laptop user that mainly use either a trackpad or a mouse to navigate on the computer. She occasionally have pain in shoulder, arms and/or wrists due to prolonged computer use. If she had the ability to control the mouse pointer by dragging her fingers on top of the keyboard like a trackpad, she might use it. She would prefer to have this area on the right side of the keyboard and that a mouse click would be done by tapping on the keyboard. She would like KeyTrack to be integrated into the keyboard and would not pay more than roughly 500 SEK.

4.3 Design

The goal with the design of the KeyTrack was to keep the design as minimal, discrete and unnoticeable as possible. To keep the KeyTrack as a part of the keyboard, with the addition of a small framework on top of the keyboard, this is in contrast to the trackpad. The trackpad is either placed beside the keyboard similarly to the mouse, or below the keyboard which is the case on most laptops. The framework consists of phototransistors and IR LEDs.

The idea was that when a finger is moving around on top of the keyboard sensors are shaded, thus registers movement. See section 4.5.7 for more information of how to find shadows.

The first design was very simple with the basic idea of phototransistors covering two or three sides of the keyboard and one side covered with IR LEDs. With a separate microcontroller covering all the input and output for the sensors and LED-lights. Below is an image showing a first draft of the prototype design, see Figure 4.1, where it is visible how the sensors were intended to be placed along the keyboard on the top part, left and right wing. The reason why they are placed on the top instead of the bottom is the concern that the light from the computer screen could potentially interfere with our phototransistors.

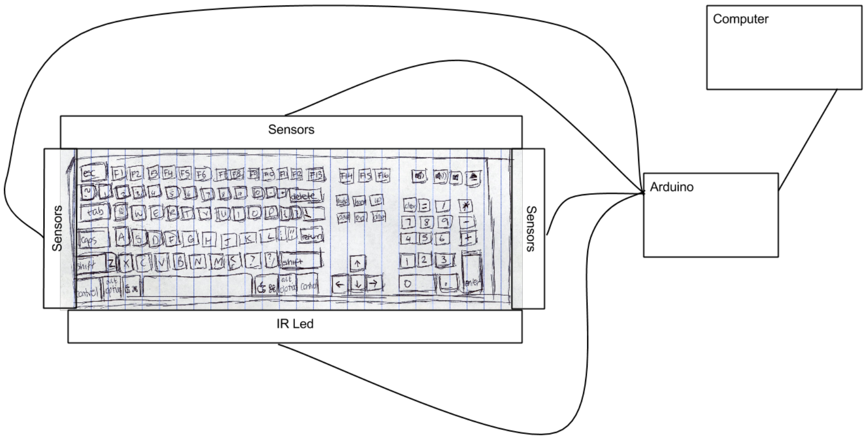


Figure 4.1: A design-draft of the prototype.

When designing the prototype, the aim was to get a high sensitivity within a minimal area. There was a restraint on how many sensors that could be used, since the size of the sensors limits how many that can be placed within a certain area. The design of the prototype was in the shape of an L, with a row of phototransistors placed on the top-right side and another on the right side of the keyboard. IR LEDs were placed on the edge of each phototransistor row. In Figure 4.2, there is a representation of how these rows are built. The IR-LEDs are visible on the sides of the prototype in Figure 4.3

It was important that the KeyTrack frame did not interfere with the users hands when typing, which was solved by placing of the KeyTrack on either the left corner or the right corner. This way, the base of the hand would never touch the frame. The decision was made to go with a simple cardboard frame design. The cardboard frame can easily be replaced and extended in the future so that a more attractive design could be developed and applied. As shown in section 2 Idea, there is an example of how a future product could look like in terms of colors and shape. There is also space for extension of the top part of the frame, by adding one more phototransistor row for a larger interaction area.

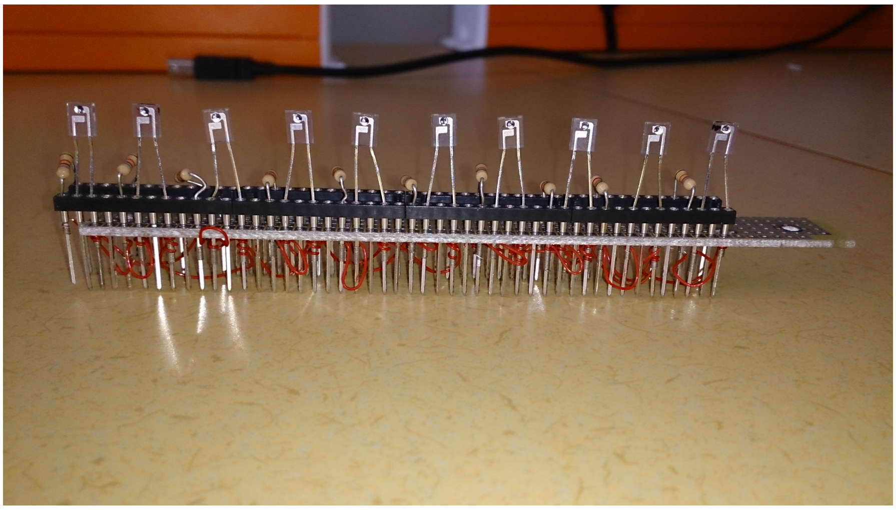


Figure 4.2: A phototransistor row.

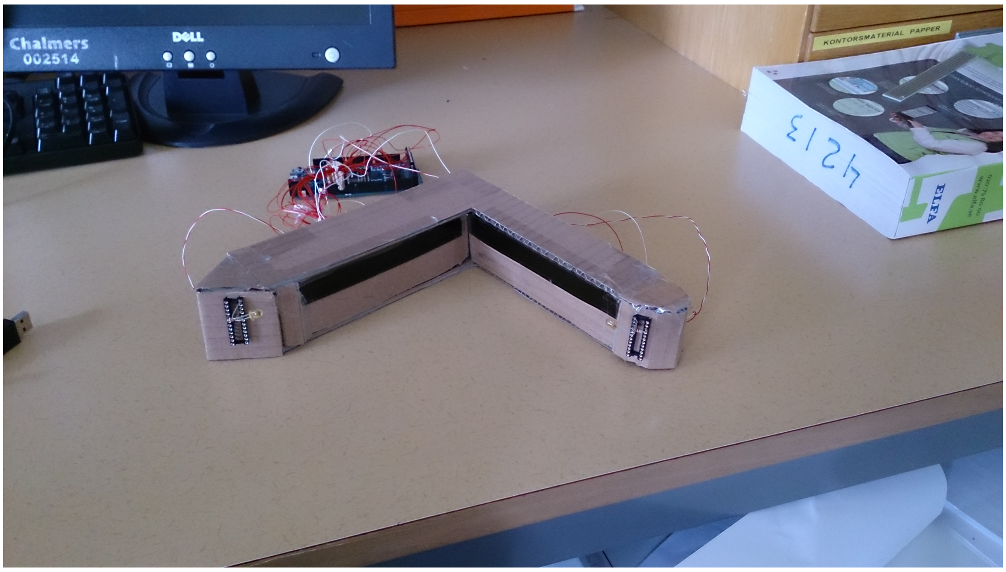


Figure 4.3: Final cardboard design prototype

The final cardboard design is 32 mm wide, 45 mm high and with a 10.9cm^2 interaction area. This is the area the phototransistors are covering. See Figure 4.3, for the final prototyp design.

4.4 Hardware

This section is about the hardware parts used in the prototype, why they were chosen and how they were implemented.

4.4.1 Arduino

Arduino is an open-source electronics platform, based on flexible and easy to use hardware. It is a way for both novices and professionals to develop prototypes of small interactive devices. The Arduino has a variety of inputs where one can connect such as sensors, lights and motors. An Arduino can either be a stand-alone project or connected to for example a PC.[28]

This project uses the Arduino Due. The Arduino Due includes functions for controlling the mouse pointer from a “native USB-port”. The Arduino is used to calculate the position of an obstruction in the IR-grid. The obstruction is supposed to be a finger meaning to control the mouse pointer on the computer.

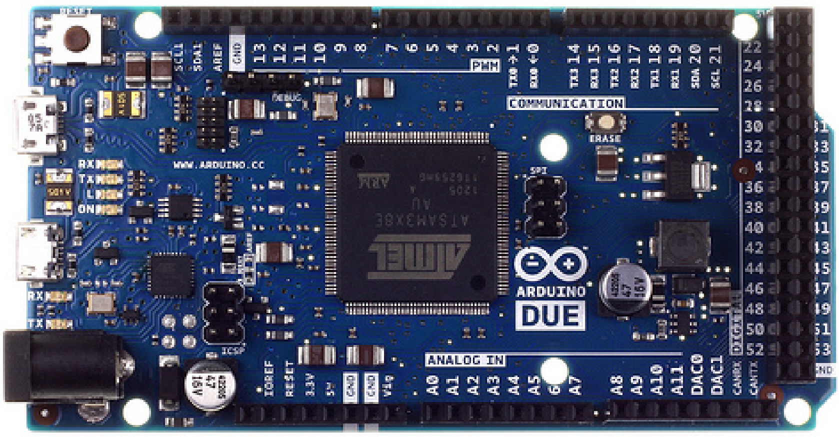


Figure 4.4: The front of an Arduino Due.[29]

Programming the Arduino The microcontrollers on the Arduino boards are programmed using the Arduino programming language based on Wiring and the Arduino development environment based on Processing. The programming language is very similar to C/C++ and can therefore be extended using C++ libraries.

4.4.2 Phototransistor

The phototransistor is a combination of a photodiode and a transistor. A simplification of the transistor would be to see it as a component that regulates the amount of current that passes through a particular circuit. The current that passes through the transistor is controlled with the transistor base, which is in turn controlled by a current. This current could be provided by for example a photodiode that generates a current when exposed to photons. By changing the current flowing through the phototransistor, the sensitivity is calibrated with the amount of photons registered. This is a property the conventional photodiode lack.[30]

In this project, phototransistors are used as receivers. The information gathered from the phototransistors are used to determine an obstruction in the IR-field. The phototransistor used are the PT908-7C manufactured by Everlight Electronics (see page 52 for a full data sheet). The phototransistor was choosen based on price and physical size.

4.4.3 IR-light

IR-light is composed of wavelengths above the spectrum that is visible to the human eye. IR lies in the range of 700 nm up to about 1 mm. IR can be used in a variety of areas such as remote temperature sensing, short-range wireless communications, night-vision and weather forecasting.[30][31]

For the project, a decision was made to use IR LEDs to provide the IR light needed for the phototransistors. This since it has many advantages compared to traditional light sources. They for example gain full efficiency in a short time and the failure rates is low and since they fade or get dimmed instead of malfunctioning. When choosing IR LEDs the goal was to match them with our sensors and still keeping costs down.[31]. See Figure 4.5, for how the phototransistor used, takes readings from different wavelengths. The reason for choosing this LED was based on the specification of the phototransistor, thus trying to match the wavelength as good as possible.

The IR LEDs found to match the purpose best, are in the wavelength 890 nm at peak emittance and are 4.9 mm wide. One important aspect is the angle of the emittance. A large emitting angle was preferable since this would decrease the need of sensors by covering a larger area. It was important that a larger angle provides light exclusively for either x or y axis. The angle of our LEDs was around 80 degrees. See Appendix C for more specifications.

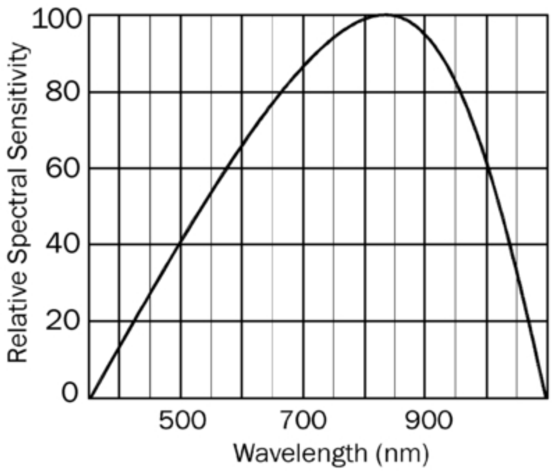


Figure 4.5: Pototransistor reading values on different wavelengths, See Appendix C for full specification

4.4.4 Resistor

A resistor is an essential component in all electronic devices. It is used to add resistance to the circuit independent of current, voltage and external factors like temperature and light conditions. There is always a risk of output anomalies when different circuits and components are connected to one another. This will in some cases result in unwanted or unexpected values while measuring the input data.

A pull-up resistor is used to ensure that a certain input stays within expected measurement levels. The resistor is called pull-up when connected to the voltage source. A pull-down resistor works basically in the same way as a pull-up resistor but is instead connected to ground. This will keep the signal near zero volts when no other active component is connected.[32]

In this project, pull-down resistors were used to make sure that the phototransistors were not giving faulty values.

4.4.5 Daylight filter

Environments where computers are normally used contain intense ambient infrared radiation. Arising from sources such as incandescent and fluorescent lamps, skylight, sunlight, and other sources[33] referred to as light noise. There was a clear problem of infrared light originating from sources which could potentially produce unwanted readings in the prototype. For instance lighting up a sensor which should be shaded.

Computer screens and television, regardless if they are LED, LCD or TFT proves to cause no problem, since the light emitted need to be visible for the human eye. Because screens need to display black areas, instead of turning the light off the solution is to either go below the visible spectrum or above. If the screen manufacturer is choosing the later of the two it could potentially cause a problem. However, no products on the market known to the group emits light above 850 nm[34].

Sunlight and fluorescent light proves to cause a problem, this is because sunlight is hard to measure since it can vary much in intensity. See Figure 4.6 for a image of sunlight, incandescent, fluorescent light, normalized power per unit. As visible there are other sources competing in the wavelength around 900 nm which are interesting. Also note that sunlight could potentially cause more readings depending on intensity.

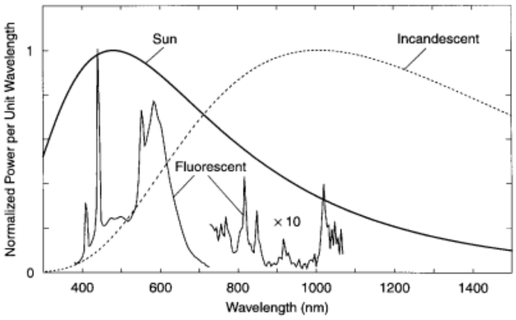


Figure 4.6: Optical power spectra of common ambient infrared sources. Spectra have been scaled to have the same maximum value.[35]

As mentioned the IR light emitters used in the project emits light with a wavelength of 890 nm. The sensors used has about 90% sensitivity at this wavelength, and reacts to light in the wavelength between 500 nm and 950 nm. They provide readings above 40% at these wavelengths. See section 4.4.3 and Figure 4.5 for a graph showing how sensitive the sensors are in different wavelengths. It is therefore required to provide a filter so that the prototype could be used in normal environments and conditions.

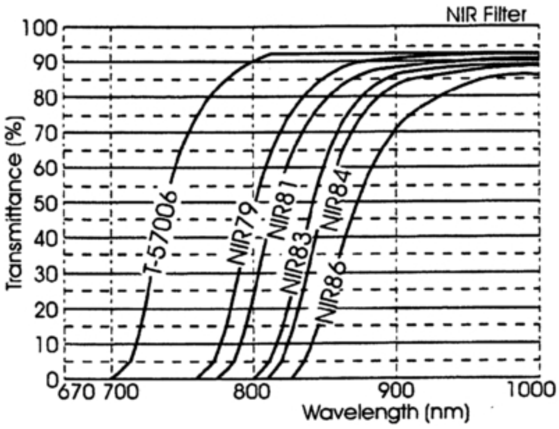


Figure 4.7: Filter NIR79 [36]

A daylight filter has a characteristic that it only allows light in a particular wavelength to pass through. As shown in Figure 4.6, there is a need to filter out the light that is not emitted from the IR LEDs. This is done by using a filter that lets wavelengths above 850 nm pass through at 90% and wavelengths below 800 nm are reduced to 50%-0% transmittance . See Figure 4.7 for transmittance on different wavelength of the filter, the filter used in the project is the NIR79.

4.4.6 Multiplexer

A multiplexer is a small component that has a number of input ports, and a smaller number of output ports (often only one). By accessing a number of dedicated control ports one can choose which of the input ports is going to be forwarded through the output port.

Multiplexers are used due to the limited number of available analog ports on the Arduino, to enable the use of more phototransistors. The multiplexers used in the project are analog multiplexers, HEF4051B manufactured by Philips Semiconductors (see page 60 for a full data sheet).

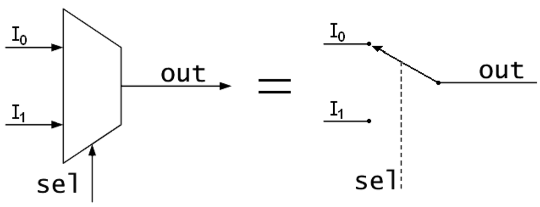


Figure 4.8: A 2 to 1 multiplexerer.[37]

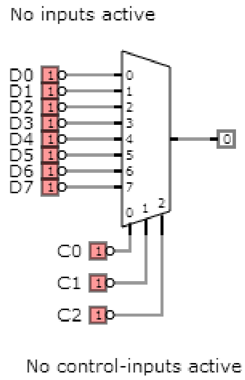


Figure 4.9: An 8 to 1 multiplexerer

4.4.7 Wire wrapping

Wire wrapping is a technique used to create prototypes of varying size without the need for soldering. Wire wrapping was very popular in the 60s and early 70s. It is a good technique that requires less skill than soldering, and is more resistant to mechanical stress than soldering points.[38]

Wire wrapping is used in this project to connect the components other than the Arduino. Every phototransistor and IR emitter is connected to a socket, placed on a stripboard and wire wrapped to make a circuit.

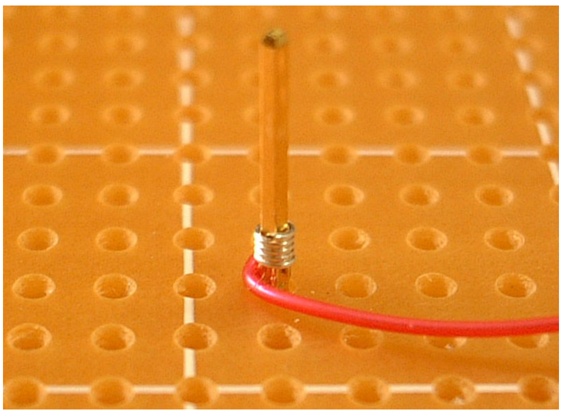


Figure 4.10: A successful wire wrap.[39]

4.5 Software

In this section, the most interesting code segments of the software will be explained in detail and the choices made about these parts will be motivated. For the full program, see appendix B.

4.5.1 Version control

Since there were multiple people writing code in parallel, version control was needed. Github was used to solve this problem. Github is a repository for software development projects that use the GIT framework for version control. Git works well with methods used in both open-source and proprietary software development projects.

4.5.2 Java GUI

A simple system tray application was written in order to give the user an ability to turn the prototype *ON* or *OFF* and to recalibrate it.

4.5.3 Arduino Mouse library

One of the primary reasons to use the Arduino Due was due to the integrated Mouse library. This library made it possible for the prototype to appear as either a mouse or a keyboard when connected to a computer. The library had two methods that was used in this project; Mouse.move() and Mouse.click().

As the name suggest, Mouse.move() was used to move the cursor in the X- And Y-axis. Mouse.click() gave the ability to simulate mouse clicks, such as left click, right click and middle click.

4.5.4 Reading sensors

To be able to read the values from the sensors in the prototype a function readSensors(), was created. This function loops through all the sensors connected to a specific multiplexer and stores the values in a global array. Since the prototype used several multiplexers, this function was called multiple times.

Parameters

The function had five parameters, all of which were integers. The first four parameters represented the different ports that the desired multiplexer was connected to. One analog port and three digital ports. The digital ports were used to choose which of the sensors connected to the multiplexer that the value should be read from. This was done by switching the digital ports between two states, “HIGH” and “LOW”, creating a binary value representing a value between zero and four. The value was then read from the analog port. The last parameter was an identifier of where the read values should be positioned in the global array.

The while loop

To acquire all the values from the sensors in the multiplexer the function had to loop through the code segment within the “while loop” five times. This because the code segment only added one of the sensor values to the list each time, and the prototype had five sensors connected to each multiplexer. The code segment within the “while loop” first sets the three digital ports to the binary value that represents the desired sensor. When this was done the value was read through the given analog port and stored as a binary value by using the function getBinValue(), at the given position in the global array of sensor values. After the value was stored, the variable “pos” was increased by one to store the next sensor value read in the next position in the list.

4.5.5 Sensor threshold

The value obtained when a sensor was read was an integer between 0 and 1023. To know if a sensor was shaded or not, a threshold value had to be computed and used to compare with the obtained sensor value. If the sensor value was lower than the threshold, it was considered as shaded. This threshold had to be computed, instead of a constant, as the prototype was moderately affected by other light sources in its vicinity. The computation was 80% of the calculated mean value of all sensor values and was executed every time the prototype was turned on or reset. The decision to only compute the threshold on startup and resets was made because computing it regularly would greatly decrease the response time.

4.5.6 Translating sensor values

The function getBinValue() was used to convert the value of a read sensor, to a binary value. This binary value represents if the sensor was shaded or not by checking if the sensor value was lower than a previously computed threshold value. The two parameters for the functions are integers that contains the value of the sensor and the threshold value.

4.5.7 Finding shadows

To be able to move the cursor in an accurate way it was needed to know exactly where the user interacts with the prototype. This was done by searching for sensors that were shaded. The function getShadows() searched for shadows on the row of sensors. A shadow was defined as one or more sensors in a row, that were shaded. If a shadow was found, the position was stored in an array with the shadows start and end position, otherwise all those values were set to a default value. In this stage the function only searched for two shadows since this was the amount needed to perform all the actions planned for the prototype. This decision was made mostly to improve the response time and make the calculations faster.

Parameters

The function had four parameters, two arrays of integers and two integers. One of the arrays consisted of the read sensor values and the other array, called shadow array, was a reference to where the shadow positions would be stored. The shadow array also stored how many shadows that were found. This was made to easily be able to check the amount of shadows found without scanning the whole array once more.

The two integers are a start and an end index number which were used to select values from the list of sensor values.

4.5.8 Moving the cursor

A function moveMouse() was implemented to translate the obtained values into a cursor movement. This function calculated the change between the current and previously read position. When the difference in X and Y direction was calculated, the function described in section 4.5.3, was used to move the cursor.

To make the movement as smooth as possible, the amount of pixels that the mouse should move was computed. This was done by computing the value obtained from the difference of the current and previous position, divided by, how long time the interaction took. A variable representing a sensitivity was also used to be able to change how fast the cursor would move.

Parameters

The parameters needed to make the calculations were four integers, the previously read position in x and y direction, and the current position in x and y direction.

4.6 Prototype iterations

The first small prototype was made using a breadboard and an Arduino Uno. The prototype included a single phototransistor and a single IR-LED. This first prototype was only able to move the cursor in one direction. A program written in Java interpreted the serial output from the Arduino and moved the mouse cursor.

The second prototype included four phototransistors and a single IR-LED. This first edition was capable of interpreting sideways movements and left mouse-clicks by using the Arduinos built-in *Mouse* library (see page 24).

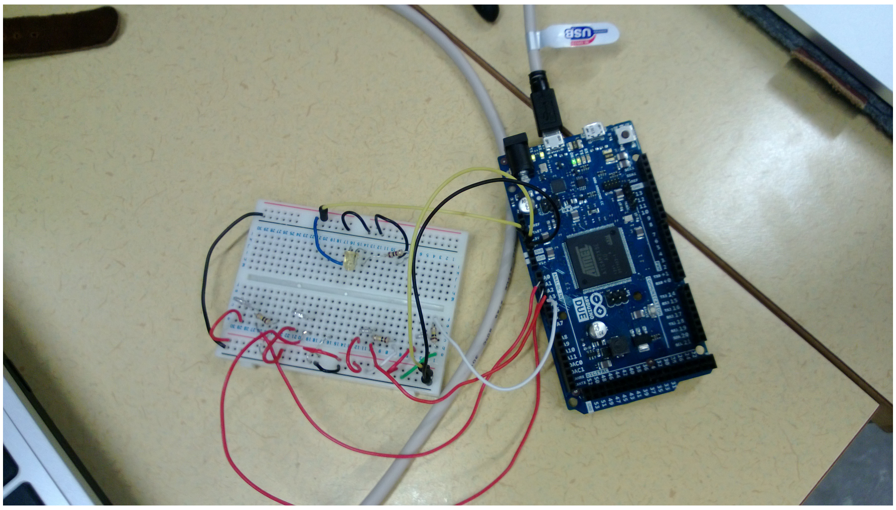


Figure 4.11: The second prototype.

Because of the limited space on the breadboard, the next prototype had to be made using some other base on which to put the hardware. Stripboards were cut into roughly 14 cm long pieces on which the hardware was put using wire-wrapping.
The computations on the Arduino Uno card was too slow. So an upgrade to the Arduino Due was made.

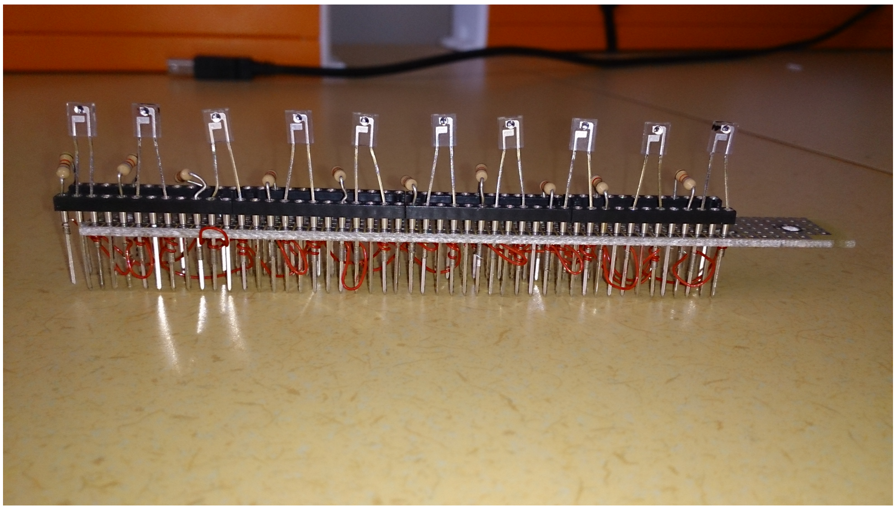


Figure 4.12: Parts of the third and latter prototypes.

5

Results

To gain an understanding of how KeyTrack reaches the levels expected within each of the previously presented requirements (see page 14), a number of tests have been made. Each test presented below were selected to give a good representation of how the KeyTrack prototype performed within each given area.

5.1 Responsiveness

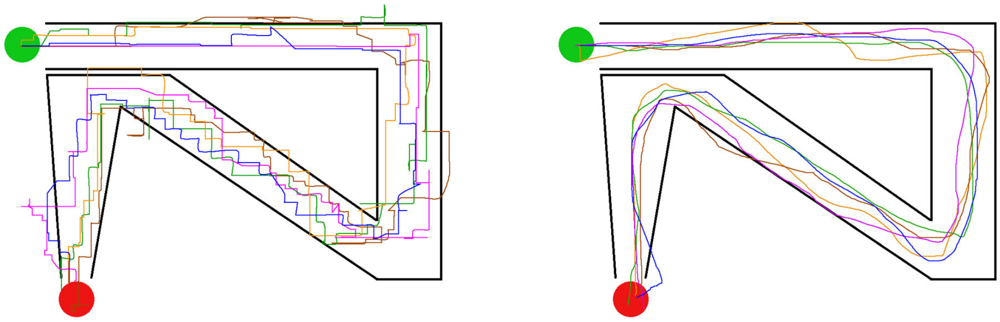
The responsiveness was measured by how long it took for a user-interaction to be translated into a cursor movement on the screen. Due to the large computation power of the Arduino Due, the prototype was very responsive. There is no measurable latency between a user interaction and cursor movement with the current setup.

5.2 Accuracy

The accuracy test was performed by a participant who was asked to move the cursor along a predefined track. The track was designed to stress-test all of the more common mouse pointer movements such as vertical, horizontal and diagonal movement. This was tested both with the prototype and with a trackpad on a *Macbook Pro*. The participant was given some time to get familiar with the prototype prior to the test and was then asked to perform the test with the following guidelines:

- Move the mouse from the green dot to the red dot as fast as possible.
- Try not to get outside of the track path.

This test was performed 5 times on each device while movement of the cursor and the time consumption was monitored. Each colored line represents one test.



(a) Prototype (b) Trackpad

Figure 5.1: Test 1 : Moving the mouse-pointer

As seen on Figure 3.1, the prototype tend to move the mouse in straight lines compared to the more soft lines of the trackpad.

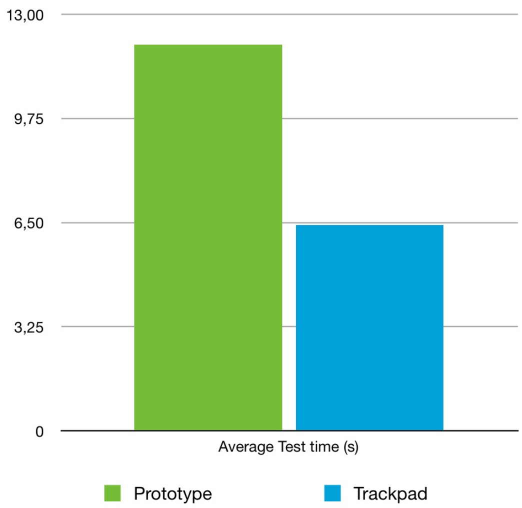


Figure 5.2: Test 1: Time results

After the test was completed, the average time it took for the participant to complete the task was calculated. The prototype took almost twice the time to complete the task, **12,06** seconds compared to **6,42** seconds.

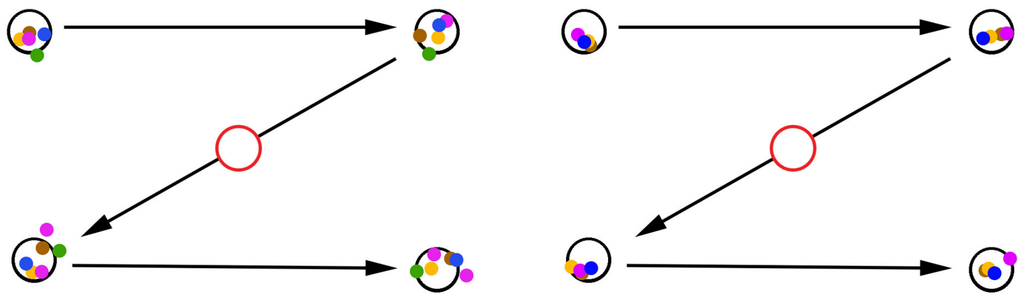
5.3 Functionality

The functionality implemented in the KeyTrack prototype is mouse movement and the ability to do left clicks. To be able to see how well left clicks were registered when using the prototype, the participant was asked to perform a simple test. The criteria for the test were:

- Navigate the cursor to each circle, in the correct order.
- Perform a left click when inside a circle.

When the user performed a left click, a dot on the screen was generated. The dot had to be inside, or at least break the line of the circle, to be approved and the user was asked to do this as fast as possible.

The user performed this test five times with the prototype and five times with a trackpad on a *Macbook Pro* for comparison. The left clicks made can be seen in Figure 5.3. Each color represents an attempt to perform the task. The red circle in the center represents the starting position of the on screen pointer for each test. The average time for all the tests on each pointing device was calculated and are presented in Figure 5.4.



(a) Prototype (b) Trackpad

Figure 5.3: Test 2 : Mouse click test

As seen in Figure 5.3 the clicks, when the prototype was used, were more scattered inside the circle which was mostly due to poor navigation. In one of the attempts made with the prototype, represented with the purple color, two accidental left clicks were registered when trying to navigate to the third and fourth circle. These can be seen outside the circles, but a dot can still be found inside since the user had to complete the task. The participant explained that this mostly was a result of stress when trying to improve the time from previous attempts. When the prototype was used the average time of the attempts were **23,68** seconds, and with the trackpad, **4,8** seconds (seen in Figure 5.4).

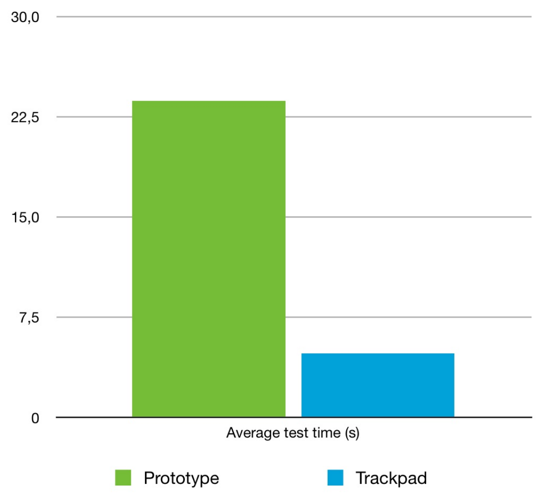
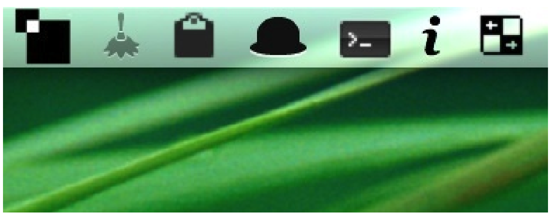


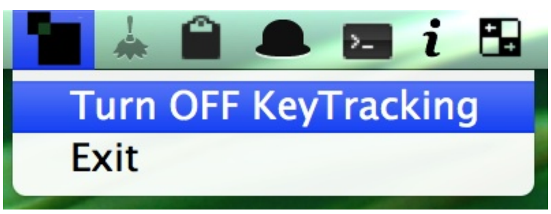
Figure 5.4: Test 2: Time results

5.4 Customization

The only customization implemented was the possibility to turn the pointing device on and off through a java application. But due to the implementation of the Mouse() library, there was a possibility to customize both sensitivity and acceleration natively. This would be done by alternating mouse settings in the operating system.



(a) Tray icon



(b) Menu

Figure 5.5: Java GUI

5.5 Final design

The current prototype of KeyTrack is sturdy and can easily be placed next to almost any keyboard.

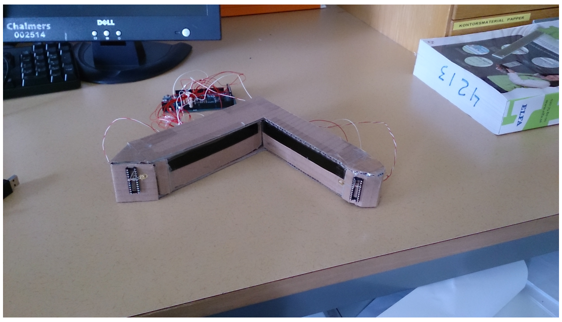


Figure 5.6: Front of KeyTrack. Daylight filter and IR light emitters visible.

It is constructed with cardboard, since it is an easily used material. Multiple wires are connected from the stripboard-mounted phototransistors (see Figure 4.12) to the Arduino's 3.3V and ground ports.

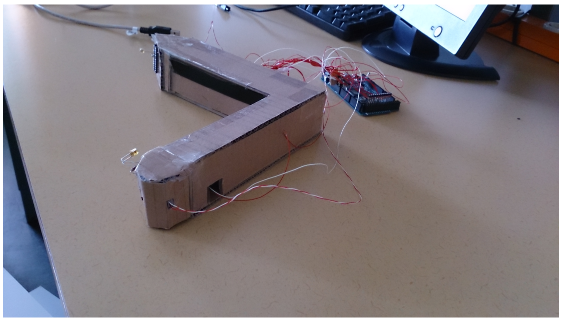


Figure 5.7: Back of KeyTrack. Wires connecting to sensors.

The multiplexers are placed on the other side of the KeyTrack and wires are connected to the Arduino's analog, digital, 5V and ground ports.

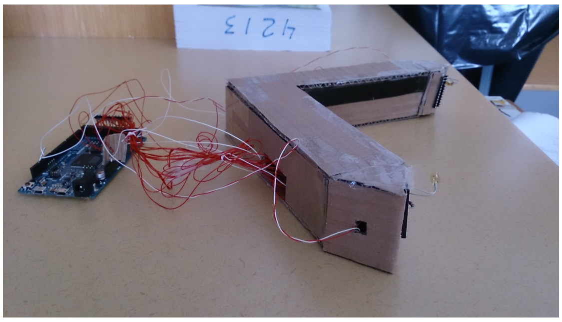


Figure 5.8: Back of KeyTrack. Wires connecting to sensors and multiplexers.

6

Discussion

The prototype developed in this project is merely a shadow of what KeyTrack could evolve into if more time was available. The results acquired from tests performed on the prototype and the general design-choices made are guidelines for possible future prototypes to be developed. In this section the results and decisions made will be discussed and analyzed.

6.1 Design

The design of the prototype does not match the design of the original concept. Even though this is the case, the prototype still gives a representation of how the KeyTrack would work and how it would improve certain aspects of computer interaction. The movement pattern needed from the user to perform tasks with a computer has been decreased since the need to reach for a separate pointing device has been eliminated. This decrease results in minimized risks for repetitive injuries.

The prototype only covers the right side of the keyboard which wasn't the original idea. A completely enveloping frame would give a much larger potential area of interaction.

The positioning of the IR emitters on the prototype are not optimal. The way they are placed only gives about half of the KeyTrack's area to interact with. Changing the IR emitters position could increase the area in which the user could control the cursor.

The choice not to build a fully enveloping frame was made because of the non-discreet nature of the prototype. The frame would make it considerably harder to type on the keyboard, and could force the user into a unnatural posture when trying to use the computer.

6.2 Requirements

The results presented previously will here discussed and put against the requirements stated on page 14.

6.2.1 Responsiveness

The responsiveness of the prototype has been very fast since the development platform was changed from an Arduino Uno to the Arduino Due. This is most likely due to the large increase in raw computational power. The only thing that actually caused the responsiveness to decrease was if the prototype self-calibrated it's sensor threshold continuously.

6.2.2 Accuracy

The prototype is working to the extent that one could say that it is somewhat of a pointing device. The performance is jumpy, resulting the pointer to randomly jump in a direction. This is caused by the algorithm that translate a shadow-movement into cursor movement. The algorithm is doing a division by time, and sometimes, the denominator is near zero (< 0.0001) due to incorrect data. This will cause the algorithm to output a large movement in a random direction.

Another issue was the grid-like movement of the mouse-pointer that was obvious in the Results chapter (see figure 5.1). The reasons of this can be narrowed down to a list of two; the Arduino card or the placement of our sensors.

The Arduino has, as previously mentioned, native USB mouse support. Since this feature is used to actually move the mouse on the computer, it could also be responsible for this grid-like movement. If it sends the movements in iterations of two, where one iteration sends an change in X-axis and the other iteration in the Y-axis, It is obvious that the cursor would move in a grid-like pattern.

Another cause could be the placement of our sensors, if they are not perfectly aligned, the shadows would be registered at different iterations. This would then result in a change in X and Y axis at different times.

6.2.3 Functionality

To navigate the cursor and left-click was the only functionality that was implemented in the prototype of KeyTrack, due to lack of time. This functionality is not sufficient to be able to compete against the traditional computer mouse. Further functionality, such as right-click and scrolling could be implemented but was not necessary for the tests done to strengthen the KeyTracks ergonomic properties. An idea has formed on how to implement further functionality, but a decision was made not to. Instead focus was placed on trying to increase the prototypes accuracy.

The result from the left click test, seen in Figure 5.3, indicates that left clicks can be executed with some accuracy but may also be performed randomly. This is an issue that needs to be addressed to make KeyTrack a viable option as a pointing device.

The functionality of KeyTrack needs to be improved, however this should be possible with more time and some alterations to the hardware and software.

6.2.4 Customization

The requirements on customization of the KeyTrack was not fulfilled by the prototype constructed in this project. The user must be allowed to change more settings on how the pointing device should behave.

7

Conclusion

The prototype has proven that integration between keyboard and mouse with infrared light is possible. The final prototype is fully functional and can be used to replace the mouse. It does, however, need more work to be a competitive alternative to the other products focused on improving ergonomics by centralizing hand movements.

To make the prototype a more competitive and a user friendly product, more functionality needs to be implemented. For instance, gestures such as scrolling, right click and other functionality possible with a regular computer mouse. This could be solved by software implementations, but due to the lack of time these implementations were not completed. Furthermore improvement in the accuracy could be made by using more phototransistors or more accurate sensors.

The IR-technique that was used for the prototype is a possible solution to the problem. The main issue with the prototype has been that other light sources have interfered with the sensors. This could be solved by matching the wavelength of sensor and IR-LEDs more precisely, and potentially use a higher wavelength to avoid light interference. A longer wavelength would avoid interference originating from common lamps.

The frame needs to have a more subtle design. The aim was to get the frame on top of the keyboard, not surrounding it. The current design should be replaced with a thinner frame that covers the whole keyboard. The final product could however be completely integrated into a keyboard, to make it more appealing.

The prototype does represent an improvement to ergonomics by involving less muscle movements and avoiding moving of spine, neck and shoulders. Thus making it a more ergonomic alternative to the regular computer mouse. Since the market for more ergonomic alternatives is increasing. There is a huge potential in the project for further improvements.

Bibliography

[1] N. R. Ball, J. N. Vardalas, Ferranti-Packard Pioneers in Canadian Electrical Manufacturing, Mcgill Queens Univ Pr, 1994.

[2] D. Engelbart, Doug engelbart 1968 demostration, Online (2009).
URL http://vimeo.com/1408300l

[3] B. Edwards, The computer mouse turns 40, Online (Dec. 2008).
URL http://www.macworld.com/article/1137400/mouse40.html

[4] A. Yassi, Repetitive strain injuries, The Lancet 349 (1997) 943–947.

[5] C. Adams, What is ergonomics, Online.
URL http://ergonomics.about.com/od/ergonomicbasics/a/ergo101.htm

[6] Leap motion product, Online.
URL https://www.leapmotion.com/product

[7] G. Franco, Ramazzini and workers' health (Sep. 1999).
URL http://155.185.2.46/immagini4/lancet354_99_858.pdf

[8] S. Jones, Computer work postures and injury: The stress of reaching for the mouse, a doctors' perspective.
URL http://www.businessknowhow.com/manage/computer-mouse-injuries.htm,read:2013-05-07

[9] Preventing rsi, Online (2010).
URL http://repetitivestraininjury.org.uk/preventing-rsi.html

[10] How to use your keyboard and mouse more efficiently, Online (2013).
URL http://www.logitech.com/en-gb/articles/528

[11] Ergonomic keyboard from microsoft.
URL http://en.wikipedia.org/wiki/File:MS_Natural_Keyboard_Pro.JPG

[12] Comfort guidelines - important ergonomic information, Online (2013).
URL `http://www.logitech.com/en-gb/articles/527`

[13] Fingers and parts of the keyboard, color-coded.
URL `http://commons.wikimedia.org/wiki/File:Typing-colour_for-finger-positions.svg`

[14] A. R. D. O. M. Camilleri, B. Chu, D. Rempel, Indirect touch pointing with desktop computing: Effects of trackpad size and input mapping on performance, posture, discomfort and preference.
URL `http://pro.sagepub.com/content/56/1/1114`

[15] Touchpad of an hp laptop.
URL `http://en.wikipedia.org/wiki/File:Touchpad_HP.JPG`

[16] Trackball (Aug. 1996).
URL `http://foldoc.org/trackball`

[17] K. Hay, Designing the leap motion controller, Online (May 2013).
URL `http://blog.leapmotion.com/post/49544844079/designing-leap-motion-controller`

[18] Montage of leapmotion pictures.
URL `http://www.squidoo.com/leap-motion-technology`

[19] Introducing the leap motion, Online (May 2013).
URL `https://www.youtube.com/watch?feature=player_embedded&v=3b4w749Tud8`

[20] Anonymous, Ergonomic rollermouse: An alternative to traditional computer mouse, Professional Safety 47 (12) (2002) 15.

[21] C. design, Rollermouse ergonomics, http://www.contour-design.co.uk/benefits/ergonomics, Read: 2013-05-10.

[22] Control your pc with your eyes - tobii pceye (2013).
URL `http://www.tobii.com/pceye`

[23] Picture of the tobiipceye.
URL `http://www.tobii.com/en/assistive-technology/global/products/hardware/pceye/`

[24] Motion sensing: Remote possibility (Feb. 2010).

[25] Nintendo, Nintendo wi (May 2013).

[26] T. K. Jiaqing Lin, Hiroaki Nishino, K. Utsumiya, Free hand interface for controlling applications based on wii remote ir sensor (Dec. 2010).

[27] E. R. G. J. A. K. Iman Habib, Niklas Berggren, M. Fjeld, Dgts: Integrated typing and pointing.
URL http://www.t2i.se/wp-content/uploads/2012/11/57270232.pdf

[28] Arduino webpage.
URL www.arduino.cc

[29] Arduino.cc, The front of an arduino due.
URL http://arduino.cc/en/Main/ArduinoBoardDue

[30] I. Poole, Phototransistor basics, Tech. rep., Electronics.

[31] A. Rogalski, Infrared Detectors, Second Edition, Taylor and Francis Group, LLC.

[32] K. Ross, Very basic circuits: Pull up resistor.
URL http://www.seattlerobotics.org/encoder/mar97/basics.html

[33] J. Kahn, J. Barry, Wireless infrared communications, Proceedings of the IEEE Volume:85 (Issue: 2) (1997) 265 – 298.

[34] E. Lueder, Liquid Crystal Displays : Addressing Schemes and Electro-Optical Effects (3rd Edition).

[35] M. D. A. R. Narasimhan, J. M. Kahn, Effect of electronic-ballast uorescent lighting on wireless infrared links, IEE Proc.-Optoelectron.

[36] Uv and ir filters, larsvenson komponent ab and Nitto Jushi Kogyo CO, LTD.
URL http://www.larsvensonkomponent.se/text1_11.html,read:2013-04-31

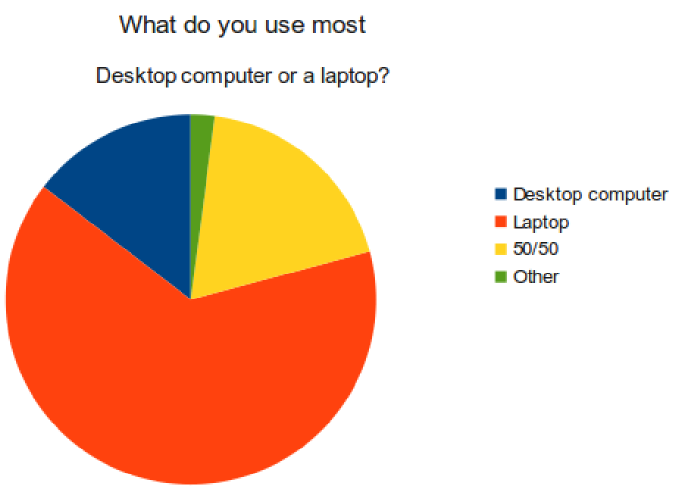
[37] Lenehey, A schematic of a 2-1 multiplexer.
URL http://en.wikipedia.org/wiki/File:Multiplexer2.png

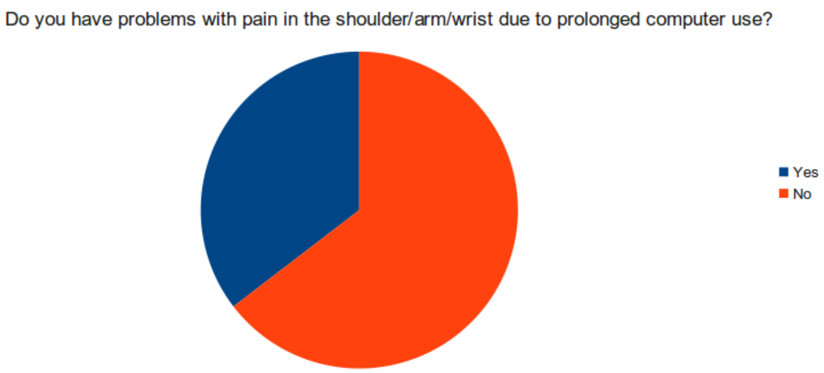
[38] M. Frank, Wire wrapping: The solderless connection.
URL http://www.tessco.com/yts/resourcecenter/pdfs/wirewrap.pdf

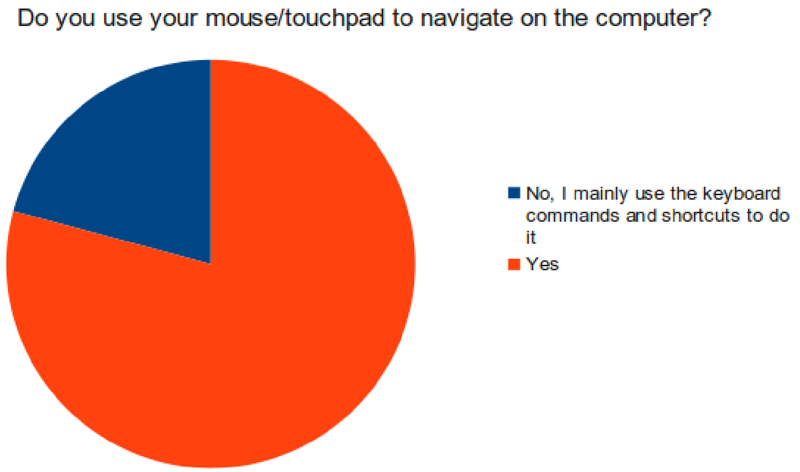
[39] Haragayato, A successful wire wrap.
URL http://en.wikipedia.org/wiki/File:Wire_Wrapping.jpg

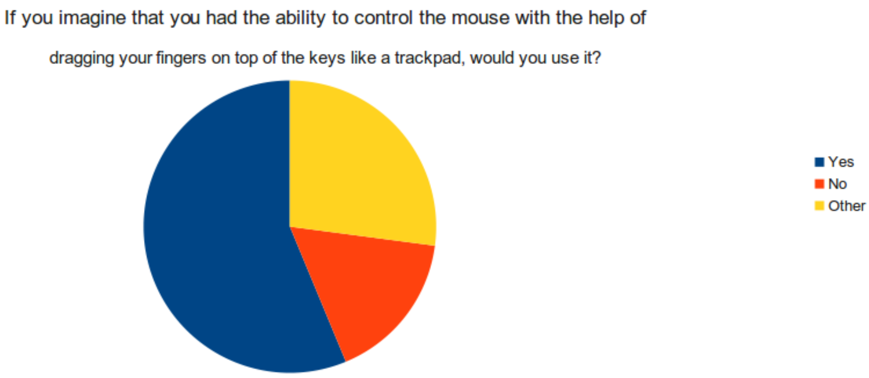
A

Survey


What do you use most
Desktop computer or a laptop?
Desktop computer
Laptop
50/50
Other


Do you have problems with pain in the shoulder/arm/wrist due to prolonged computer use?
Yes
No


Do you use your mouse/touchpad to navigate on the computer?
No, I mainly use the keyboard commands and shortcuts to do it
Yes


If you imagine that you had the ability to control the mouse with the help of dragging your fingers on top of the keys like a trackpad, would you use it?
Yes
No
Other

B

Source code

```
#include "RunningAverage.h"

//Variables

boolean DEV_MODE = false;


float currentTime = millis();
double prevX = -1;
double currX = -1;

double prevY = -1;
double currY = -1;

float clickTime = millis();
double prevClickX = -1;
double prevClickY = -1;

boolean isClick = false;


//DEFINE AND ALLOCATE #OFSENSRORS
int sensArray[20];
int sensValue[20];

int X_START_POS = 0;
int X_END_POS = 9;
int Y_START_POS = 10;
int Y_END_POS = 19;

float prevXTime = millis();
float prevYTime = millis();

//Digital MUX THINGS
```

```
int dig1 = 30;
int dig2 = 32;
int dig3 = 34;

int dig4 = 36;
int dig5 = 38;
int dig6 = 40;

int dig7 = 42;
int dig8 = 44;
int dig9 = 46;

int dig10 = 48;
int dig11 = 50;
int dig12 = 52;

int a0 = 0;
int a1 = 1;
int a2 = 2;
int a3 = 3;

//Array to read data from mux input 0 - 4
int digOut[15] = {
  0, 0, 0,
  1, 0, 0,
  0, 1, 0,
  1, 1, 0,
  0, 0, 1};

//DEFINE AVERAGE RULES
const int NR_OF_AVERAGE_MEASUREMENTS = 2;
RunningAverage averageX(10 * NR_OF_AVERAGE_MEASUREMENTS);
RunningAverage averageY(10 * NR_OF_AVERAGE_MEASUREMENTS);

//CALIBRATION THRESHOLDS
int THRESHOLD_X = 0;
int THRESHOLD_Y = 0;

const double SENSITIVITY = 4.2;

//SHADOW ARRAY
int shadowXArray[5] = {
  -1,-1,-1,-1,0};
  int shadowYArray[5] = {
    -1,-1,-1,-1,0};



    void setup() {
      Serial.begin(9600);
      Mouse.begin();
```

```
//Init Digital Pinmodes
pinMode(dig1, OUTPUT);
pinMode(dig2, OUTPUT);
pinMode(dig3, OUTPUT);
pinMode(dig4, OUTPUT);
pinMode(dig5, OUTPUT);
pinMode(dig6, OUTPUT);
pinMode(dig7, OUTPUT);
pinMode(dig8, OUTPUT);
pinMode(dig9, OUTPUT);
pinMode(dig10, OUTPUT);
pinMode(dig11, OUTPUT);
pinMode(dig12, OUTPUT);

//Setup Average Class
averageX.clear();
averageY.clear();

readAnalogSensors(a0, dig10, dig11, dig12, 0);
readAnalogSensors(a1, dig7, dig8, dig9,5);
readAnalogSensors(a2, dig4, dig5, dig6,10);
readAnalogSensors(a3, dig1, dig2, dig3,15);

calibrateSensors();

Serial.println("Start");
}

void loop() {
  currentTime = millis();


//readSensor();
readSensors(a0, dig10, dig11, dig12, 0);
readSensors(a1, dig7, dig8, dig9, 5);
readSensors(a2, dig4, dig5, dig6, 10);
readSensors(a3, dig1, dig2, dig3, 15);

//calibrate sensor Threshold
//calibrateSensors();

//Search for shadows
getShadows(sensArray, X_START_POS, X_END_POS, shadowXArray);
getShadows(sensArray, Y_START_POS, Y_END_POS, shadowYArray);

//Store X and Y position
currX = (shadowXArray[0] + shadowXArray[1])/2;
currY = (shadowYArray[0] + shadowYArray[1])/2;

//move on-screen pointer
moveMouse(prevX, currX, prevY, currY);

//check if the interaction is a click
```

```
isClick = doClick(prevX, currX);

//Store curr X and Y as prev
prevX = currX;
prevY = currY;

if(DEV_MODE){
  readAnalogSensors(a0, dig10, dig11, dig12, 0);
  readAnalogSensors(a1, dig7, dig8, dig9,5);
  readAnalogSensors(a2, dig4, dig5, dig6,10);
  readAnalogSensors(a3, dig1, dig2, dig3,15);
  printSensArrayValues();
  delay(500);
}
}

void allocateArray(int array[], int length){
  for(int i = 0; i < length; i++){
    array[i] = 1;
  }
}

//Translate sensor values to bin value
int getBinValue(int value, int sVal){
  int binValue;
  if (value < sVal){
    binValue = 0;
  }
  else{
    binValue = 1;
  }
  return binValue;
}

boolean doClick(int pX, int cX){
  if(pX == -1 && cX != -1){
    clickTime = currentTime;
    prevClickX = cX;
    return true;
  }
  else if(isClick == true && abs(prevClickX-pX) <= 1
    && abs(clickTime - currentTime < 200 && cX != -1)){
    return true;
  }
  else if(isClick == true && cX == -1){
    Mouse.click();
    return false;
  }
  else return false;
}

void getShadows(int array[], int startPos, int endPos, int *shadowRef){
  int nrOfShadows = 0;
```

```
  for(int i = startPos; i<=endPos; i++){
    if(array[i] == 0){ // Om skugga hittas

      shadowRef[nrOfShadows*2] = i - startPos; //L gg till skuggans
          startposition

      for(int j = i; j<=endPos-1; j++){
        if(array[j+1] == 1 || j+1>=endPos){ //kolla om skuggan  r  slut
          shadowRef[nrOfShadows*2+1] = j - startPos; //L gg till skuggan
              slutposition
          i = j+1;
          break;
        }
      }
      nrOfShadows++;
    }
  }
  if(nrOfShadows>2 || nrOfShadows == 0){
    shadowRef[0] = -1;
    shadowRef[1] = -1;
    shadowRef[2] = -1;
    shadowRef[3] = -1;
    shadowRef[4] = 0;
  }
  else{
    shadowRef[4] = nrOfShadows; //L gg till skuggan slutposition
    }
}

void moveMouse(int pX, int cX, int pY, int cY){
  double diffX = cX - pX;
  double diffY= cY - pY;
  int moveX = 0;
  int moveY = 0;

  if(pX != -1 && cX != -1 && diffX != 0 ){
    moveX = diffX*SENSITIVITY/(((currentTime - prevXTime)/200));
    prevXTime = currentTime;
  }
  if(pY != -1 && cY != -1 && diffY != 0 ){
    moveY = diffY*SENSITIVITY/(((currentTime - prevYTime)/200));
    prevYTime = currentTime;
  }
  if(abs(moveX) < 30 && abs(moveY) < 30){
    Mouse.move(moveX, moveY, 0);
  }
}

  void readSensors(int analog, int d1, int d2, int d3, int pos){
    int step = 0;
    while(step < 15){
      digitalWrite(d1, digOut[step]);
      step++;
```

```
    digitalWrite(d2, digOut[step]);
    step++;
    digitalWrite(d3, digOut[step]);
    step++;
    sensArray[pos] = getBinValue(analogRead(analog), THRESHOLD_X);
    pos++;
  }
}

void readAnalogSensors(int analog, int d1, int d2, int d3, int pos){
  int step = 0;
  while(step < 15){
    digitalWrite(d1, digOut[step]);
    step++;
    digitalWrite(d2, digOut[step]);
    step++;
    digitalWrite(d3, digOut[step]);
    step++;
    sensValue[pos] = analogRead(analog);
    pos++;
  }
}



void calibrateSensors(){
  for(int i = X_START_POS; i <= X_END_POS && X_END_POS != 0; i++){
    averageX.addValue(sensValue[i]);
  }

  for(int i = Y_START_POS ; i <= Y_END_POS && Y_END_POS != 0; i++) {
    averageY.addValue(sensValue[i]);
  }

  THRESHOLD_X = averageX.getAverage()*0.8;
  THRESHOLD_Y = averageY.getAverage()*0.8;

}

void printSensArrayValues(){
  Serial.println("_________________________");
  Serial.println("INPUTVALUES: ");
  for(int i = 0;i<20;i++){
    Serial.print(sensValue[i]);
    Serial.print(" ");
  }
  Serial.println();
  Serial.println("_________________________");
  Serial.println("SENS ARRAY: ");

  Serial.println("X Values: ");
  for(int i = X_START_POS; i <= X_END_POS && X_END_POS != 0; i++){
```

```
        Serial.print(sensArray[i]);
        Serial.print(" ");
      }
      Serial.println("");

      Serial.println("Y Values: ");
      for(int i = Y_START_POS ; i <= Y_END_POS && Y_END_POS != 0; i++) {
        Serial.print(sensArray[i]);
        Serial.print(" ");
      }
      Serial.println("_________________________");

      Serial.println("_________________________");
      Serial.println("SHADOW ARRAYS: ");

      Serial.println("X shadows: ");
      for(int i = 0; i <= 4 ; i++){
        Serial.print(shadowXArray[i]);
        Serial.print(" ");
      }
      Serial.println("");

      Serial.println("Y shadows: ");
      for(int i = 0; i <= 4 ; i++){
        Serial.print(shadowYArray[i]);
        Serial.print(" ");
      }
      Serial.println("");

      Serial.println("_________________________");

      Serial.print("THRESHOLD X: ");
      Serial.println(THRESHOLD_X);
      Serial.print("THRESHOLD Y: ");
      Serial.println(THRESHOLD_Y);

    }

}
```

C

Phototransistors - Datasheet

Technical Data Sheet

1.5mm Side Looking Phototransistor

PT908-7C-F

Features

- Fast response time
- High sensitivity
- Small junction capacitance
- Pb Free
- This product itself will remain within RoHS compliant version.

Descriptions

PT908-7C-F is a phototransistor in miniature package which is molded in a water clear plastic with spherical top view lens. The device is spectrally matched to infrared emitting diode.

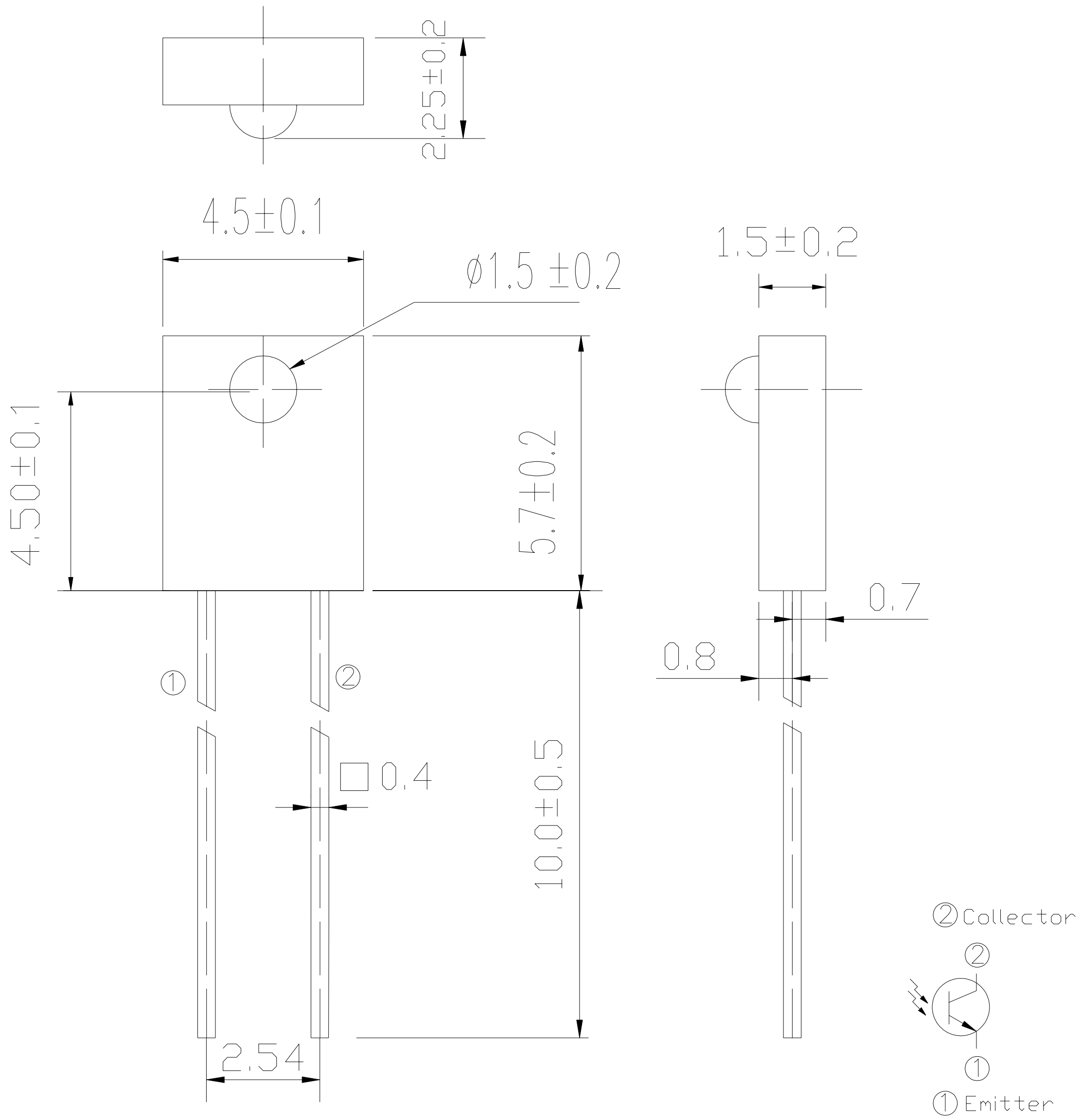
Applications

- Optoelectronic switch
- VCR，Video Camera
- Floppy disk drive
- Infrared applied system

Device Selection Guide

LED Part No.	Chip Material	Lens Color
PT908-7C-F	Silicon	Water Clear

Package Dimensions



Notes: 1.All dimensions are in millimeters

2.Tolerances unless dimensions ±0.15mm

Absolute Maximum Ratings (Ta=25℃)

Parameter	Symbol	Rating	Units
Collector-Emitter Voltage	V_{CEO}	30	V
Emitter-Collector-Voltage	V_{ECO}	5	V
Collector Current	I_C	20	mA
Operating Temperature	Topr	-25 ~ +85℃	℃
Storage Temperature	Tstg	-40 ~ +100℃	℃
Lead Soldering Temperature	Tsol	260	℃
Power Dissipation at (or below) 25℃ Free Air Temperature	PD	75	mW

Notes: *1:Soldering time≦5 seconds.

Electro-Optical Characteristics (Ta=25℃)

Parameter	Symbol	Condition	Min.	Typ.	Max.	Units
Collector – Emitter Breakdown Voltage	BV_{CEO}	I_C=100μA Ee=0mW/cm^2	30	---	---	V
Emitter-Collector Breakdown Voltage	BV_{ECO}	I_E=100μA Ee=0mW/cm^2	5	---	---	V
Collector-Emitter Saturation Voltage	$V_{CE)(sat)}$	I_C=2mA Ee=1mW/cm^2	---	---	0.4	V
Rise Time	t_r	V_{CE}=5V I_C=1mA RL=1000Ω	---	15	---	μS
Fall Time	t_f		---	15	---	
Collector Dark Current	I_{CEO}	Ee=0mW/cm^2 V_{CE}=20V	---	---	100	nA
On State Collector Current	$I_{C(on)}$	Ee=0.555mW/cm^2 V_{CE}=5V	0.80		5.0	mA
Wavelength of Peak Sensitivity	λp	---	---	940	---	nm
Rang of Spectral Bandwidth	$\lambda_{0.5}$	---	400	---	1100	nm

PT908-7C-F

Typical Electro-Optical Characteristics Curves

Fig.1 Collector Power Dissipation vs. Ambient Temperature

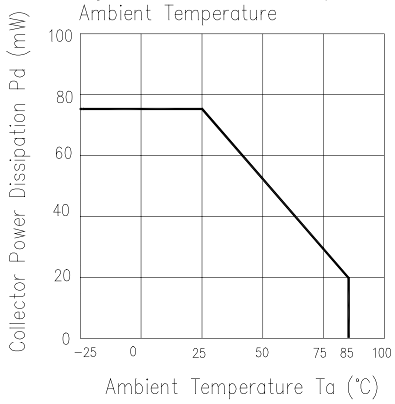


Fig.2 Collector Dark Current vs.Ambient Temperature

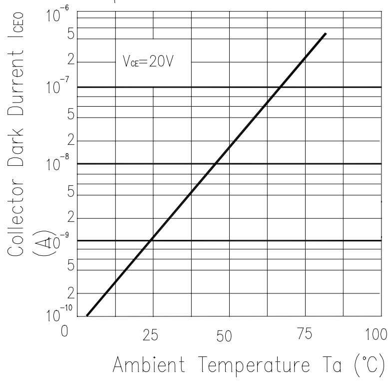


Fig.3 Relative Collector Current vs. Ambient Temperature

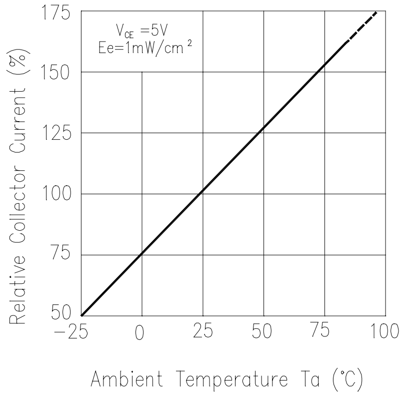


Fig.4 Collector Current vs. Irradiance

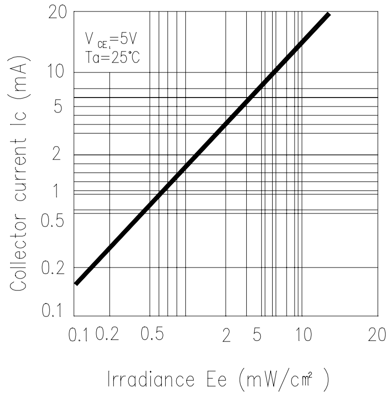


Fig.5 Spectral Sensitivity

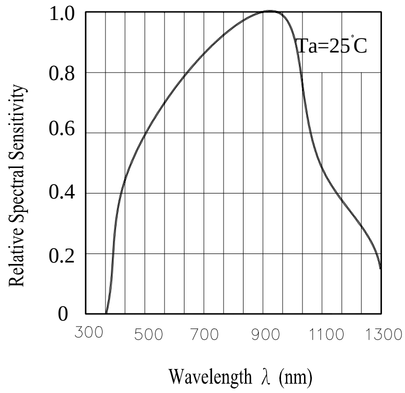
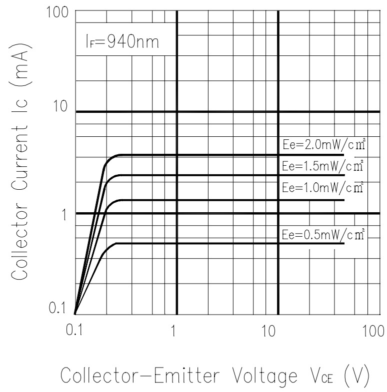
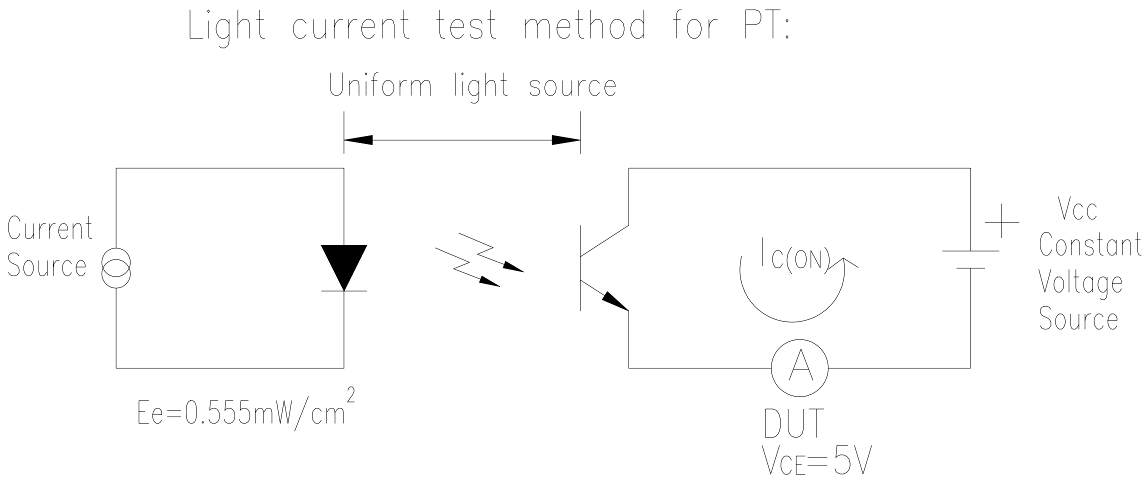


Fig.6 Collector Current vs. Collector Current Ic (mA)



Test method



Ranks

Parameter	Symbol	Min	Max	Unit	Test condition
BIN1	$I_{C(ON)}$	0.80	1.53	mA	V_{CE}=5V Ee=0.555mW/cm^2
BIN2		1.11	1.98		
BIN3		1.43	2.68		
BIN4		1.59	3.06		
BIN5		2.0	5.0		

Reliability Test Item And Condition

The reliability of products shall be satisfied with items listed below.

Confidence level：90%

LTPD：10%

NO.	Item	Test Conditions	Test Hours/ Cycles	Sample Sizes	Failure Judgement Criteria	Ac/Re
1	Solder Heat	TEMP:260℃±5℃	10sec	22pcs	More than 90% of lead to be covered by soldering	0/1
2	Temperature Cycle	H : +100℃ 15mins ↕ 5mins L : -40℃ 15mins	300Cycle	22pcs		0/1
3	Thermal Shock	H :+100℃ 5mins ↕ 10secs L :-10℃ 5mins	300Cycle	22pcs	I_R≧U×2 Ee≦L×0.8 V_F≧U×1.2	0/1
4	High Temperature Storage	TEMP.：+100℃	1000hrs	22pcs	U：Upper Specification Limit	0/1
5	Low Temperature Storage	TEMP.：-40℃	1000hrs	22pcs	L：Lower Specification Limit	0/1
6	DC Operating Life	V_{CE}=5V	1000hrs	22pcs		0/1
7	High Temperature/ High Humidity	85℃ / 85% R.H	1000hrs	22pcs		0/1

Packing Quantity Specification

1. 1000 Pcs/1Bag，10 Bags/1Box
2. 10 Boxes/1Carton

Label Form Specification

PN: Customer’s Production Number
P/N : Production Number
QTY: Packing Quantity
CAT: Ranks
HUE: Peak Wavelength
REF: Reference
LOT No: Lot Number

Notes

1. Above specification may be changed without notice. EVERLIGHT will reserve authority on material change for above specification.
2. When using this product, please observe the absolute maximum ratings and the instructions for using outlined in these specification sheets. EVERLIGHT assumes no responsibility for any damage resulting from use of the product which does not comply with the absolute maximum ratings and the instructions included in these specification sheets.
3. These specification sheets include materials protected under copyright of EVERLIGHT corporation. Please don’t reproduce or cause anyone to reproduce them without EVERLIGHT’s consent.

EVERLIGHT ELECTRONICS CO., LTD.
Office: No 25, Lane 76, Sec 3, Chung Yang Rd, Tucheng, Taipei 236, Taiwan, R.O.C
Tel: 886-2-2267-2000, 2267-9936
Fax: 886-2267-6244, 2267-6189, 2267-6306
http:\\www.everlight.com

D

Multiplexer - Datasheet

DATA SHEET

For a complete data sheet, please also download:

- The IC04 LOCMOS HE4000B Logic Family Specifications HEF, HEC
- The IC04 LOCMOS HE4000B Logic Package Outlines/Information HEF, HEC

HEF4051B
MSI
8-channel analogue multiplexer/demultiplexer

Product specification
File under Integrated Circuits, IC04

January 1995

Philips Semiconductors

PHILIPS

DESCRIPTION

The HEF4051B is an 8-channel analogue multiplexer/demultiplexer with three address inputs (A_0 to A_2), an active LOW enable input ($\overline{E}$), eight independent inputs/outputs (Y_0 to Y_7) and a common input/output (Z).

The device contains eight bidirectional analogue switches, each with one side connected to an independent input/output (Y_0 to Y_7) and the other side connected to a common input/output (Z).

With $\overline{E}$ LOW, one of the eight switches is selected (low impedance ON-state) by A_0 to A_2. With $\overline{E}$ HIGH, all switches are in the high impedance OFF-state, independent of A_0 to A_2.

V_{DD} and V_{SS} are the supply voltage connections for the digital control inputs (A_0 to A_2, and $\overline{E}$).
The V_{DD} to V_{SS} range is 3 to 15 V.

The analogue inputs/outputs (Y_0 to Y_7, and Z) can swing between V_{DD} as a positive limit and V_{EE} as a negative limit. V_{DD}–V_{EE} may not exceed 15 V.

For operation as a digital multiplexer/demultiplexer, V_{EE} is connected to V_{SS} (typically ground).

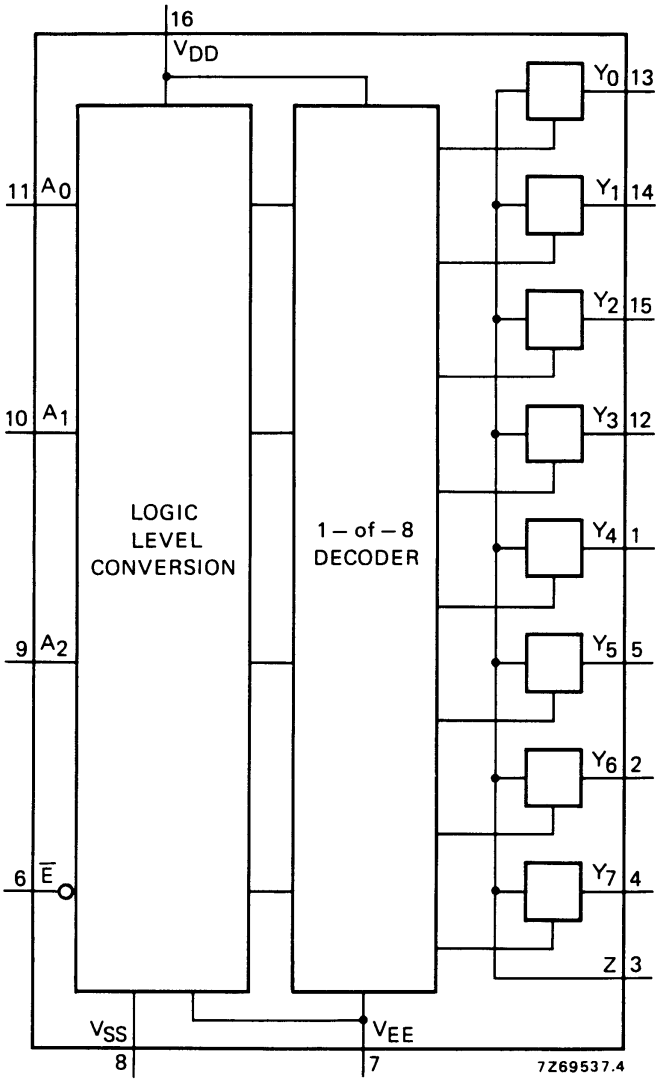


Fig.1 Functional diagram.

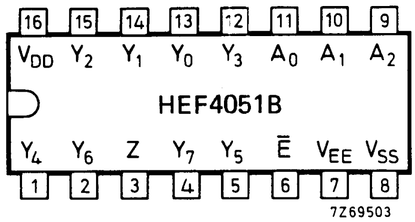


Fig.2 Pinning diagram.

HEF4051BP(N):	16-lead DIL; plastic (SOT38-1)
HEF4051BD(F):	16-lead DIL; ceramic (cerdip) (SOT74)
HEF4051BT(D):	16-lead SO; plastic (SOT109-1)

( ): Package Designator North America

PINNING

Y_0 to Y_7	independent inputs/outputs
A_0 to A_2	address inputs
$\overline{E}$	enable input (active LOW)
Z	common input/output

FAMILY DATA, I_{DD} LIMITS category MSI

See Family Specifications.

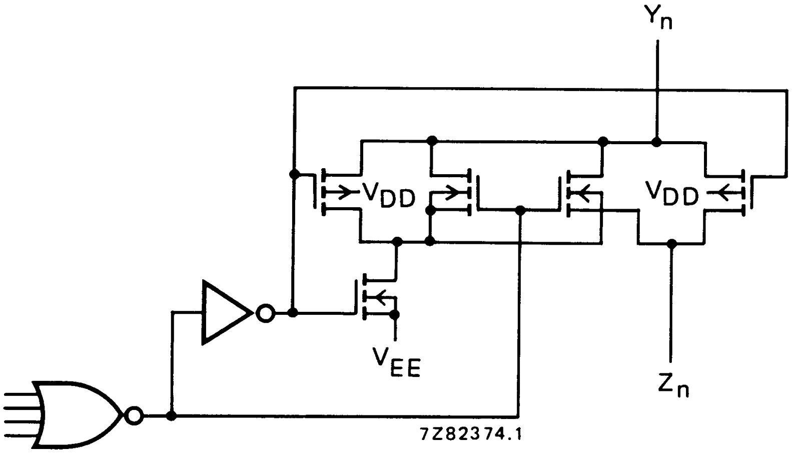


Fig.3 Schematic diagram (one switch).

FUNCTION TABLE

INPUTS				CHANNEL ON
$\overline{E}$	A_2	A_1	A_0	
L	L	L	L	Y_0–Z
L	L	L	H	Y_1–Z
L	L	H	L	Y_2–Z
L	L	H	H	Y_3–Z
L	H	L	L	Y_4–Z
L	H	L	H	Y_5–Z
L	H	H	L	Y_6–Z
L	H	H	H	Y_7–Z
H	X	X	X	none

Notes

1. H = HIGH state (the more positive voltage)
   L = LOW state (the less positive voltage)
   X = state is immaterial

RATINGS

Limiting values in accordance with the Absolute Maximum System (IEC 134)

Supply voltage (with reference to V_{DD})	V_{EE}	–18 to + 0,5 V

Note

1. To avoid drawing V_{DD} current out of terminal Z, when switch current flows into terminals Y, the voltage drop across the bidirectional switch must not exceed 0,4 V. If the switch current flows into terminal Z, no V_{DD} current will flow out of terminals Y, in this case there is no limit for the voltage drop across the switch, but the voltages at Y and Z may not exceed V_{DD} or V_{EE}.

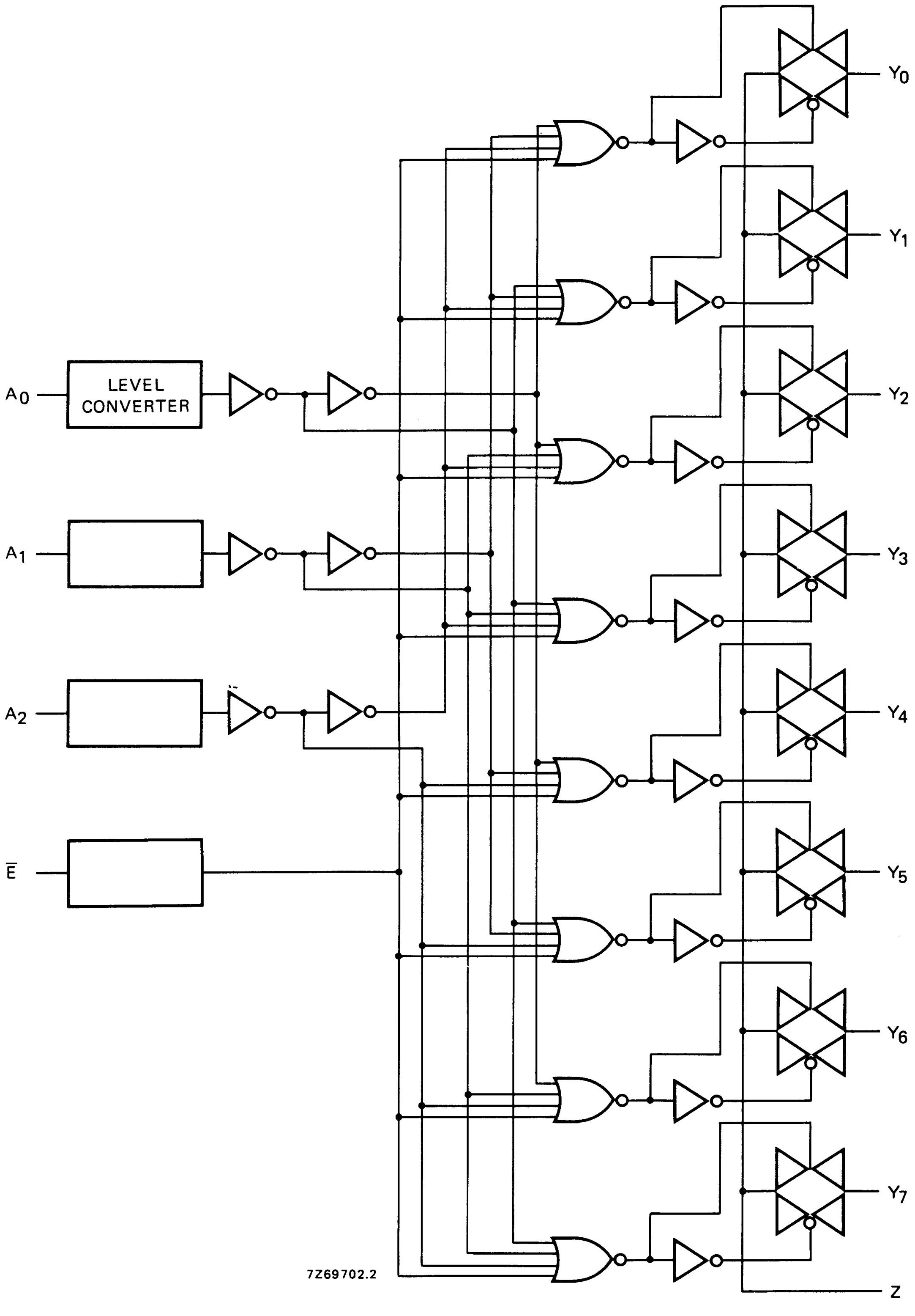


Fig.4 Logic diagram.

DC CHARACTERISTICS

T_{amb} = 25 °C

	$V_{DD}-V_{EE}$ V	SYMBOL	TYP.	MAX.		CONDITIONS
ON resistance	5 10 15	R_{ON}	350 80 60	2500 245 175	Ω Ω Ω	V_{is} = 0 to $V_{DD}-V_{EE}$ see Fig.6
ON resistance	5 10 15	R_{ON}	115 50 40	340 160 115	Ω Ω Ω	V_{is} = 0 see Fig.6
ON resistance	5 10 15	R_{ON}	120 65 50	365 200 155	Ω Ω Ω	V_{is} = $V_{DD}-V_{EE}$ see Fig.6
‘Δ’ ON resistance between any two channels	5 10 15	ΔR_{ON}	25 10 5	– – –	Ω Ω Ω	V_{is} = 0 to $V_{DD}-V_{EE}$ see Fig.6
OFF-state leakage current, all channels OFF	5 10 15	I_{OZZ}	– – –	– – 1000	nA nA nA	$\overline{E}$ at V_{DD} $V_{SS} = V_{EE}$
OFF-state leakage current, any channel	5 10 15	I_{OZY}	– – –	– – 200	nA nA nA	$\overline{E}$ at V_{SS} $V_{SS} = V_{EE}$

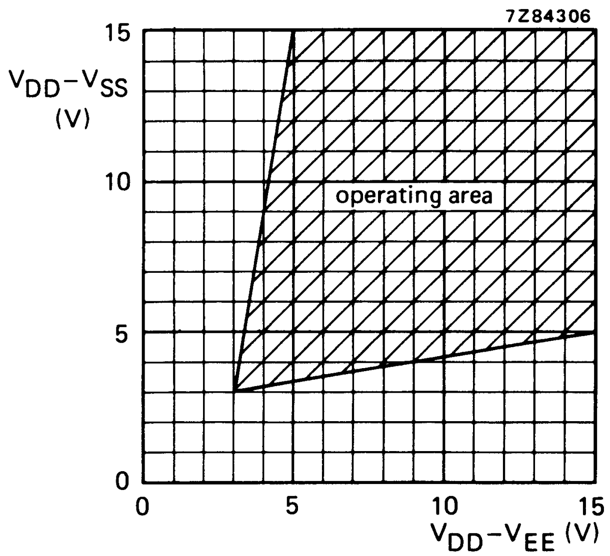


Fig.5 Operating area as a function of the supply voltages.

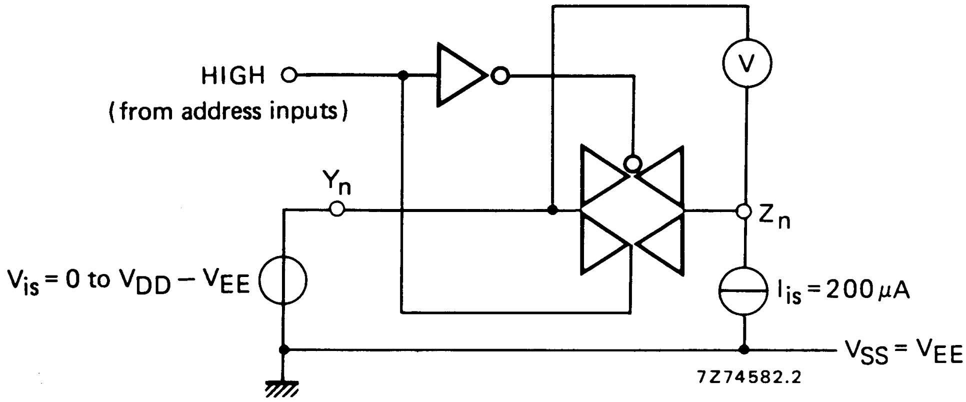
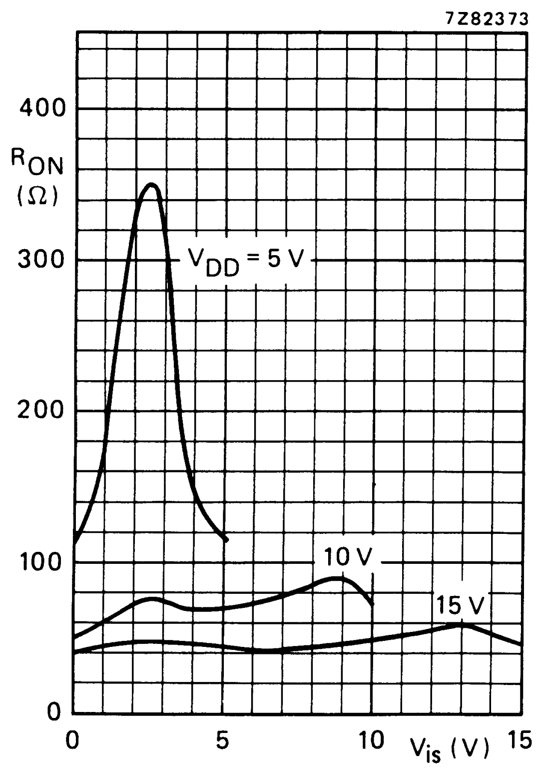


Fig.6 Test set-up for measuring R_{ON}.



I_{is} = 200 μA
$V_{SS} = V_{EE}$ = 0 V

Fig.7 Typical R_{ON} as a function of input voltage.

AC CHARACTERISTICS

$V_{EE} = V_{SS} = 0$ V; $T_{amb} = 25$ °C; input transition times ≤ 20 ns

	V_{DD} V	TYPICAL FORMULA FOR P (μW)	
Dynamic power dissipation per package (P)	5	$1\,000\ f_i + \sum(f_oC_L) \times V_{DD}^2$	where
	10	$5\,500\ f_i + \sum(f_oC_L) \times V_{DD}^2$	f_i = input freq. (MHz)
	15	$15\,000\ f_i + \sum(f_oC_L) \times V_{DD}^2$	f_o = output freq. (MHz)
			C_L = load capacitance (pF)
			$\sum(f_oC_L)$ = sum of outputs
			V_{DD} = supply voltage (V)

AC CHARACTERISTICS

$V_{EE} = V_{SS} = 0$ V; $T_{amb} = 25$ °C; input transition times ≤ 20 ns

	V_{DD} V	SYMBOL	TYP.	MAX.		
Propagation delays						
$V_{is} \rightarrow V_{os}$	5		15	30	ns	
HIGH to LOW	10	t_{PHL}	5	10	ns	note 1
	15		5	10	ns	
	5		15	30	ns	
LOW to HIGH	10	t_{PLH}	5	10	ns	note 1
	15		5	10	ns	
$A_n \rightarrow V_{os}$	5		150	300	ns	
HIGH to LOW	10	t_{PHL}	60	120	ns	note 2
	15		45	90	ns	
	5		150	300	ns	
LOW to HIGH	10	t_{PLH}	65	130	ns	note 2
	15		45	90	ns	
Output disable times						
$\overline{E} \rightarrow V_{os}$	5		120	240	ns	
HIGH	10	t_{PHZ}	90	180	ns	note 3
	15		85	170	ns	
	5		145	290	ns	
LOW	10	t_{PLZ}	120	240	ns	note 3
	15		115	230	ns	
Output enable times						
$\overline{E} \rightarrow V_{os}$	5		140	280	ns	
HIGH	10	t_{PZH}	55	110	ns	note 3
	15		40	80	ns	
	5		140	280	ns	
LOW	10	t_{PZL}	55	110	ns	note 3
	15		40	80	ns	

	V_{DD} V	SYMBOL	TYP.	MAX.		
Distortion, sine-wave response	5		0,25		%	
	10		0,04		%	note 4
	15		0,04		%	
Crosstalk between any two channels	5		–		MHz	
	10		1		MHz	note 5
	15		–		MHz	
Crosstalk; enable or address input to output	5		–		mV	
	10		50		mV	note 6
	15		–		mV	
OFF-state feed-through	5		–		MHz	
	10		1		MHz	note 7
	15		–		MHz	
ON-state frequency response	5		13		MHz	
	10		40		MHz	note 8
	15		70		MHz	

Notes

V_{is} is the input voltage at a Y or Z terminal, whichever is assigned as input.

V_{os} is the output voltage at a Y or Z terminal, whichever is assigned as output.

1. R_L = 10 kΩ to V_{EE}; C_L = 50 pF to V_{EE}; $\overline{E}$ = V_{SS}; V_{is} = V_{DD} (square-wave); see Fig.8.
2. R_L = 10 kΩ; C_L = 50 pF to V_{EE}; $\overline{E}$ = V_{SS}; A_n = V_{DD} (square-wave); V_{is} = V_{DD} and R_L to V_{EE} for t_{PLH}; V_{is} = V_{EE} and R_L to V_{DD} for t_{PHL}; see Fig.8.
3. R_L = 10 kΩ; C_L = 50 pF to V_{EE}; $\overline{E}$ = V_{DD} (square-wave);
   V_{is} = V_{DD} and R_L to V_{EE} for t_{PHZ} and t_{PZH};
   V_{is} = V_{EE} and R_L to V_{DD} for t_{PLZ} and t_{PZL}; see Fig.8.
4. R_L = 10 kΩ; C_L = 15 pF; channel ON; V_{is} = ½ $V_{DD\ (p\text{-}p)}$ (sine-wave, symmetrical about ½ V_{DD}); f_{is} = 1 kHz; seeFig.9.
5. R_L = 1 kΩ; V_{is} = ½ $V_{DD\ (p\text{-}p)}$ (sine-wave, symmetrical about ½ V_{DD});

   $$20\log\frac{V_{os}}{V_{is}} = -50\text{ dB; see Fig. 10.}$$

6. R_L = 10 kΩ to V_{EE}; C_L = 15 pF to V_{EE}; $\overline{E}$ or A_n = V_{DD} (square-wave); crosstalk is $|V_{os}|$(peak value); see Fig.8.
7. R_L = 1 kΩ; C_L = 5 pF; channel OFF; V_{is} = ½ $V_{DD\ (p\text{-}p)}$ (sine-wave, symmetrical about ½ V_{DD});

   $$20\log\frac{V_{os}}{V_{is}} = -50\text{ dB; see Fig. 9.}$$

8. R_L = 1 kΩ; C_L = 5 pF; channel ON; V_{is} = ½ $V_{DD\ (p\text{-}p)}$ (sine-wave, symmetrical about ½ V_{DD});

   $$20\log\frac{V_{os}}{V_{is}} = -3\text{ dB; see Fig. 9.}$$

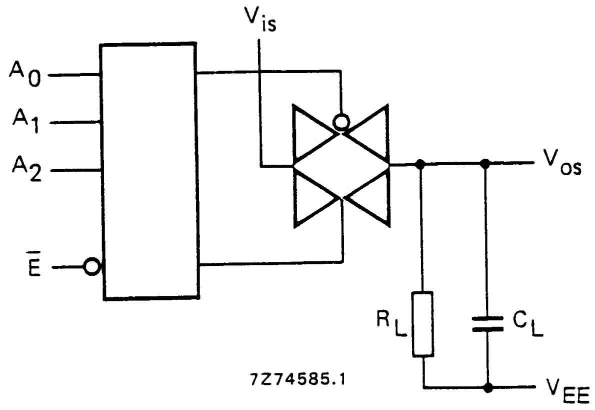


Fig.8

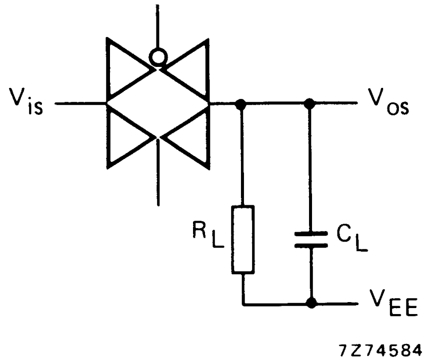
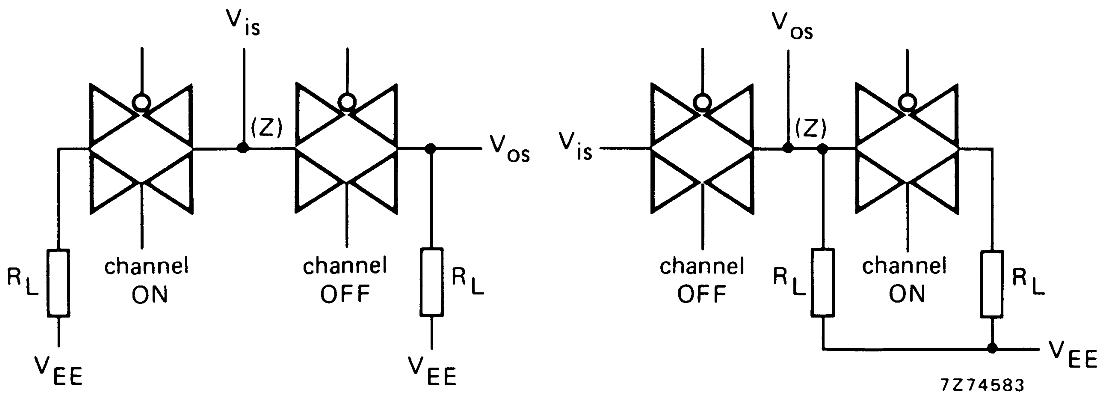


Fig.9



(a)

(b)

Fig.10

APPLICATION INFORMATION

Some examples of applications for the HEF4051B are:

- Analogue multiplexing and demultiplexing.
- Digital multiplexing and demultiplexing.
- Signal gating.

NOTE

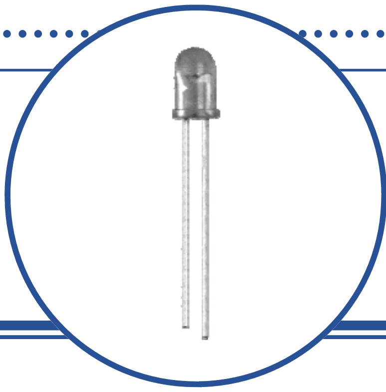
If break before make is needed, then it is necessary to use the enable input.

E

IR emitters - Datasheet

Plastic Infrared Emitting Diode

OP290 Series



Features:

- Choice of narrow or wide irradiance pattern
- Choice of power ranges
- Choice of T-1¾, TO-18 or T-46 package
- Higher power output than GaAs at equivalent LEDs

Description:

Each device in this series, is a gallium aluminum arsenide infrared Light Emitting Diode (LED) that is molded in an IR-transmissive package with a wavelength centered at 890 nm, which closely matches the spectral response of silicon phototransistors, except for OP298 (AA, AB, AC, AD), which has either an 850 nm or 875 nm center wavelength. For identification purposes, each LED anode lead is longer than the cathode lead. ***Package T-1¾*** devices include: **OP290**, **OP291**, **OP292**, **OP294, OP295, OP296**, **OP297**, **OP299 (A, B, C) and OP297FAB**, ***Plastic Package TO-18*** *or* ***TO-46*** devices include: **OP293** and **OP298 (A, B, C, AA, AB, AC, AD)**.

Each **OP290, OP291** and **OP292** series come in three electrical parameters options A, B and C. The **OP290** series forward current is specified under pulse conditions up to 1.5 amps, the **OP291** series forward current is specified under pulse conditions up to 100 milliamps and the **OP292** series forward current is specified under pulse conditions up to 1 amp. The Cathode Lead length is 0.06" (1.52 mm) shorter than the Anode Lead. The silver-copper lead frame offers excellent thermal characteristics.

Each OP293 and **OP298** series come in three electrical parameter options A, B and C. The **OP293** series has an included emission angle of 60° while the **OP298** series has an included emission angle of 25°. The Cathode Lead length is 0.06" (1.52 mm) shorter than the Anode Lead. These devices, which come in a variety of power ranges offering a low cost replacement for TO-18 or TO-46 hermetic packages.

Each OP298 series come with a high irradiance output versions with four electrical parameter options AA, AB, AC and AD. These power options are in the range of **5X** greater than the A, B or C options. The **OP298** series has an included emission angle of 25°. The Cathode Lead length is 0.06" (1.52 mm) shorter than the Anode Lead. These devices, which come in a variety of power ranges offering a low cost replacement for TO-18 or TO-46 hermetic packages.

OP294 and **OP299** are designed for low-current or power-limited applications, such as battery supplies. They are similar to the **OP290** and **OP295**, but use a smaller chip that increases output efficiency at low current levels by increasing current density. Light output can be maximized with continuous (D.C.) forward current up to 100 mA or with pulsed forward current up to 750 mA. The Cathode Lead length is 0.06" (1.52 mm) shorter than the Anode Lead.

Each **OP295, OP296** and **OP297** series come in three electrical parameters options A, B and C. The **OP295** series forward current is specified under pulse conditions up to 5 amps, the **OP296** series forward current is specified under pulse conditions up to 2 amps and the **OP297** series forward current is specified under pulse conditions up to 1 amp. The Cathode Lead length is 0.06" (1.52 mm) shorter than the Anode Lead. The **OP297FAB** has a reversed polarity from the **OP297A, B** or **C**. The silver-copper lead frame offers excellent thermal characteristics.

All of these devices are spectrally and mechanically matched to the OP593 and OP598 series phototransistors.

Please refer to Application Bulletins 208 and 210 for additional design information and reliability (degradation) data.

Applications:

- Non-contact reflective object sensor
- Assembly line automation
- Machine automation
- Machine safety
- End of travel sensor
- Door sensor
- Battery-operated applications

RoHS

Part Number Guide — OP290 - OP299 Series

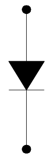
OP 2 9 X X W

Optek Assembly

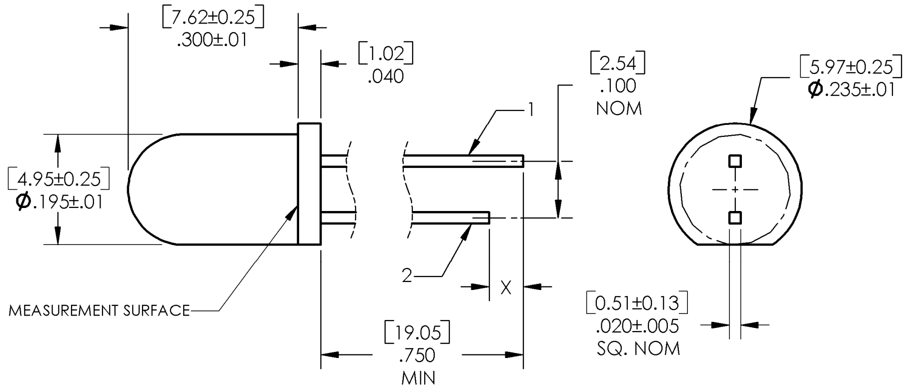
Photodiode Output Family

Maximum Forward Current
- 0, 5 — 5 amps
- 1, 6 — 2 amps
- 2, 7 — 1 amps
- 3, 8 — 200 milli-amps
- 4, 9 — 750 milli-amps

Electrical Specification Variations:
- A — Parameter A
- B — Parameter B
- C — Parameter C
- D — Parameter D
- AA — Parameter BA
- AB — Parameter BB
- AC — Parameter BC
- AD — Parameter BD



T-1¾ Package
OP290, OP291, OP292, OP294, OP295, OP296, OP297, OP299



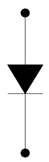
Electrical Connection A, B, C, D

Pin #	LED X=0.060" (1.52 mm)
1	Anode
2	Cathode

Electrical Connection OP297AB

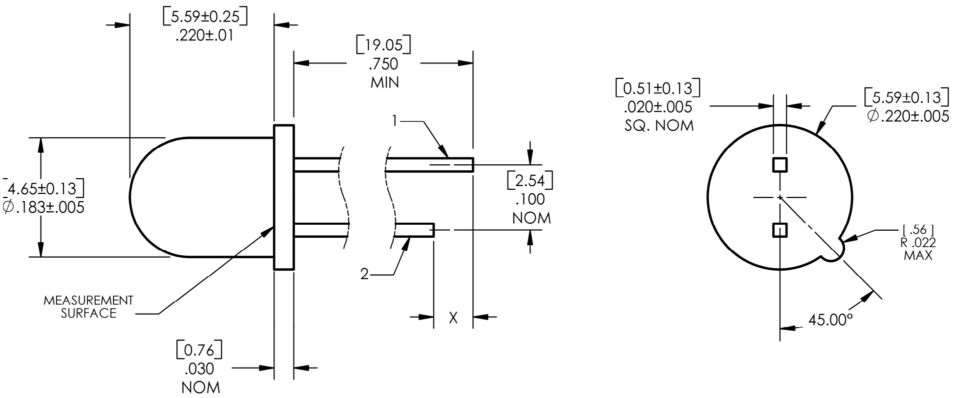
Pin #	LED X=0.060" (1.52 mm)
1	Cathode
2	Anode

TO-18, TO-46 Package
OP293 & OP298



Electrical Connection
A, B, C, AA, AB, AC, AD

Pin #	LED X=0.060" (1.52 mm)
1	Anode
2	Cathode



DIMENSIONS ARE IN: **[MILLIMETERS] INCHES**

OPTEK reserves the right to make changes at any time in order to improve design and to supply the best product possible.

Absolute Maximum Ratings (T_A=25°C unless otherwise noted)

Storage and Operating Temperature Range	-40° C to +100° C
Reverse Voltage OP290, OP292, OP294, OP295, OP297, OP299 OP291, OP293, OP296, OP298	 5.0 V 2.0 V
Continuous Forward Current OP290, OP291, OP292 OP294, OP295, OP299 OP295, OP296, OP297	 150 mA(1) 100 mA(1) 150 mA(1)
Continuous Forward Current, OP293, OP298 Free Air Board Mounted Full Heat Sink	 100 mA 133 mA 200 mA
Peak Forward Current OP290, OP295 (25 µs pulse width) OP291, OP296 (100 µs pulse width) OP292, OP297 (100 µs pulse width) OP293, OP298 (25 µs pulse width) OP294, OP299	 5.0 A 2.0 A 1.00 A 2.0 A 750 mA

Notes:
1. For OP290, OP291, OP292, OP295, OP296 and OP297, derate linearly 1.67 mA/° C above 25° C (free-air). When used with heat sink (see note 5), derate linearly 2.07 mA/° C above 65° C (normal use). For OP293 and OP298, when measured in free-air, derate power dissipation linearly 1.43 mW/° C above 25° C. For OP294 and OP299, derate linearly 1.80 mW/° C above 25° C.

Absolute Maximum Ratings (T_A=25°C unless otherwise noted)

Maximum Duty Cycle OP290 (25 µs pulse width @ 5 A)	 1.25%(1)
Lead Soldering Temperature [1/16 inch (1.6 mm) from case for 5 seconds with soldering iron]	260° C(2)
Power Dissipation, Free Air OP290, OP291, OP292, OP295, OP296, OP297 OP293, OP298 Power Dissipation, Board Mounted OP290, OP291, OP292, op295, OP296, OP297 OP293, OP298 Power Dissipation, Full Heat Sink OP290, OP291, OP292, OP295, OP296, OP297 OP293, OP298	 333 mW(3) 142 mW(3) 533 mW(4) 200 mW(4) 1.11 W(5) 400 mW(5)
Power Dissipation OP294, OP299	 180 mW

Notes:
1. For OP290, OP291, OP292, OP295, OP296 and OP297, refer to graph of Maximum Peak Pulse Current vs Pulse Width.
2. For all OPs in this series, RMA flux is recommended. Duration can be extended to 10 second maximum when soldering. A maximum of 20 grams force may be applied to the leads when flow soldering.
3. For OP290, OP291, OP292, OP295, OP296 and OP297, measured in free-air. Derate linearly 3.33 mW/° C above 25° C.
4. For OP290, OP291and OP292, mounted on 1/16" (1.6 mm) thick PCBoard with each lead soldered through 80 mil square lands 0.250" (6.35 mm) below flange of device. Derate linearly 5.33 mW/°C above 62.5°. For OP293 and OP298, mounted on 1/16" (1.60 mm) thick PCBoard with each lead soldered through 80 mil square lands 0.250" (6.35 mm) below flange of device. Derate power dissipation linearly 2.00 mW/°C above 25° C (normal use). For OP295, OP296 and OP297, mounted on 1/16" (1.6 mm) thick PCBoard with each lead soldered through 80 mil square lands 0.250" (6.35 mm) below flange of device. Derate linearly 5.33 mW/°C above 25° C.
5. Immersed in silicone fluid to simulate infinite heat sink. For OP290, OP291 and OP292, derate linearly 11.1 mW/°C above 95°C. For OP293 and OP298, derate power dissipation linearly 2.50 mW/° C above 25° C. For OP295, OP296 and OP297, derate linearly 11.1 mW/° C above 25° C.

Electrical Characteristics (T_A = 25°C unless otherwise noted)

Input Diode

SYMBOL	PARAMETER	MIN	TYP	MAX	UNITS	TEST CONDITIONS
$E_{E\,(APT)}$ (2)	Apertured Radiant Incidence OP290A OP290B OP290C	210 180 150	- - -	- 300 -	mW/cm²	I_F = 1.50 A (1)(2) Measured into a 0.250" [6.35mm] aperture 0.2" (5.08 mm) from the tip of the lens.
	OP291A OP291B OP291C	16 13 10	- - -	- 26 -		I_F = 100 mA (1)(2) Measured into a 0.250" [6.35mm] aperture 0.2" (5.08 mm) from the tip of the lens.
	OP292A OP292B OP292C	2.7 2.2 1.7	- 3.6 -	- 4.4 -		I_F = 20 mA (1)(2) Measured into a 0.250" [6.35mm] aperture 0.2" (5.08 mm) from the tip of the lens.
	OP293A OP293B OP293C	16 13 10	- 22 -	- 26 -		I_F = 100 mA (1)(2) Measured into a 0.250" [6.35mm] aperture 0.2" (5.08 mm) from the tip of the lens.
	OP294	0.50	-	1.50		I_F = 5 mA (1)(2) Measured into a 0.250" [6.35mm] aperture 0.200" (5.08mm) from the tip of the lens.
	OP295A OP295B OP295C	44 33 22	- - -	- 77 -		I_F = 1.50 A (1)(2) Measured into a 0.250" [6.35mm] aperture 1.129" (28.7 mm) from the tip of the lens.
	OP296A OP296B OP296C	3.6 2.6 1.6	- - -	- 6.6 -		I_F = 100 mA (1)(2) Measured into a 0.250" [6.35mm] aperture 1.129" (28.7 mm) from the tip of the lens.
	OP297FAB OP297A OP297B OP297C	2.4 0.7 0.5 0.3	- - 1.0 -	- - 1.3 -		I_F = 20 mA (1)(2) Measured into a 0.250" [6.35mm] aperture 1.129" (28.7 mm) from the tip of the lens.
	OP298A OP298B OP298C	3.0 2.4 1.8	- - -	- 4.8 -		I_F = 100 mA (1)(2) Measured into a 0.250" [6.35mm] aperture 0.2" (5.08 mm) from the tip of the lens.
	OP298AA OP298AB OP298AC OP298AD	3.5 3.5 6.5 8.5	- - - -	- 8.5 11.5 -		I_F = 100 mA (1)(2) Measured into a 0.250" [6.35mm] aperture 1.129" (28.7 mm) from the tip of the lens.
	OP299	0.15	-	0.45		I_F = 100 mA (1)(2) Measured into a 0.250" [6.35mm] aperture 1.129" (28.7 mm) from the tip of the lens.

Notes:
1. Measurement is taken at the end of a single 100 µs pulse. Heating due to increased pulse rate or pulse width will cause a decrease in reading.
2. Measurement of the average apertured radiant energy incident upon a sensing area 0.250" (6.35 mm) in diameter perpendicular to and centered on the mechanical axis of the lens and the specified distance from the end of the device. On all models in this series, $E_{E(APT)}$ is not necessarily uniform within the measured area.
3. Measurement is taken at the end of a single 10 ms pulse. Heating due to increased pulse rate or pulse width will cause a decrease in reading.

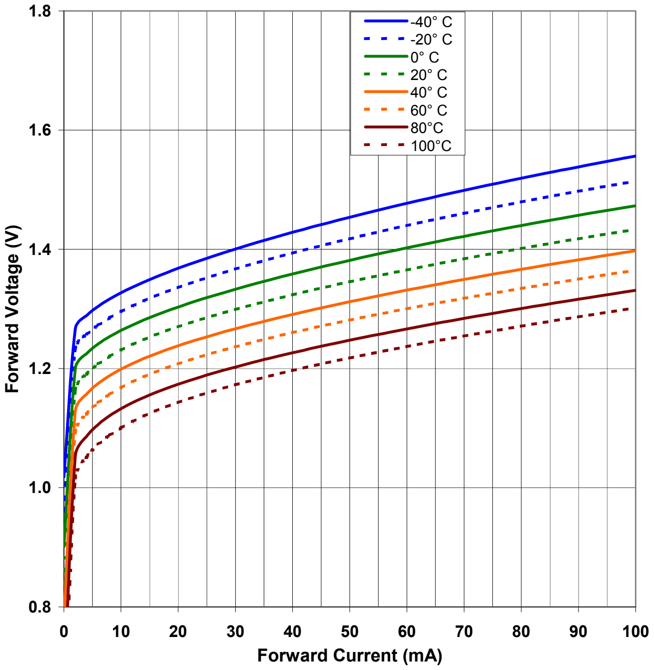
OPTEK Technology Inc. — 1645 Wallace Drive, Carrollton, Texas 75006
Phone: (972) 323-2200 or (800) 341-4747 FAX: (972) 323-2396 sensors@optekinc.com www.optekinc.com

Electrical Characteristics (T_A = 25°C unless otherwise noted)

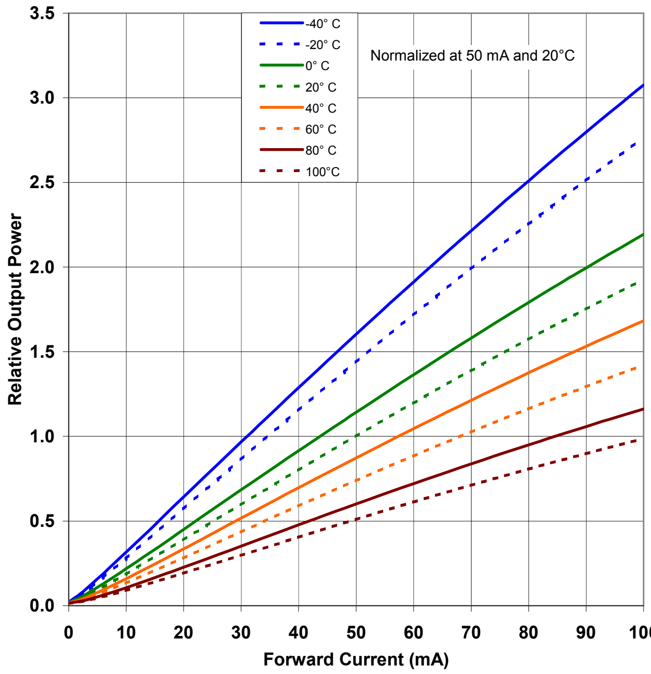
SYMBOL	PARAMETER	MIN	TYP	MAX	UNITS	TEST CONDITIONS
Input Diode						
V_F	Forward Voltage[3] OP290, OP295 OP291, OP296 OP292, OP297, OP297FAB OP293, OP298 (A, B, C) OP298 (AA, AB, AC, AD) OP294, OP299	- - - - - -	- - - - - -	4.00 2.00 1.75 2.00 2.00 1.50	V	I_F = 1.50 A I_F = 100 mA I_F = 20 mA I_F = 1.50 A I_F = 100 mA I_F = 5 mA
I_R	Reverse Current[3] OP290, OP292 OP291, OP293, OP298 (A, B, C), OP296 OP298 (AA, AB, AC, AD) OP294, OP299 OP295, OP297 OP297FAB	- - - - - -	- - - - - -	10 100 100 10 10 15	µA	V_R= 5 V V_R= 2 V V_R= 2 V V_R= 2 V V_R= 5 V V_R= 5 V
λ_P	Wavelength at Peak Emission OP290, OP291, OP292, OP293, OP294, OP295, OP296, OP297, OP298 (A, B, C), OP299 OP297FAB, OP298 (AA, AB, AC, AD)	- -	890 875	- -	nm	I_F = 10 mA
B	Spectral Bandwidth between Half Power Points	-	80	-	nm	I_F = 10 mA
$\Delta\lambda_P / \Delta T$	Spectral Shift with Temperature	-	+0.18	-	nm/°C	I_F = Constant
θ_{HP}	Emission Angle at Half Power Points OP290, OP291, OP292, OP294 OP293 OP295, OP296, OP297, OP299 OP298	- - - -	50 60 20 25	- - - -	Degree	I_F = 20 mA
t_r	Output Rise Time	-	500	-	ns	$I_{F(PK)}$=100 mA, PW=10 µs, and D.C.=10.0%
t_f	Output Fall Time	-	250	-	ns	

OP290, OP291, OP292, OP293, OP294, (A, C)

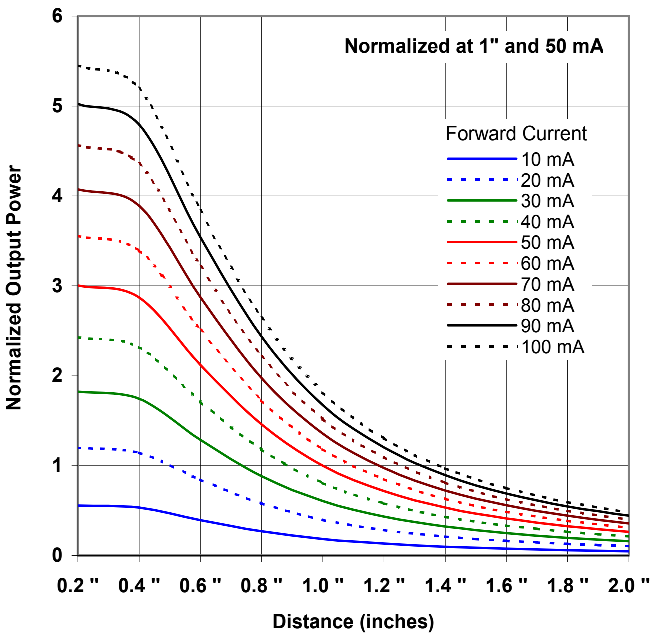
Forward Voltage vs Forward Current vs Temp.



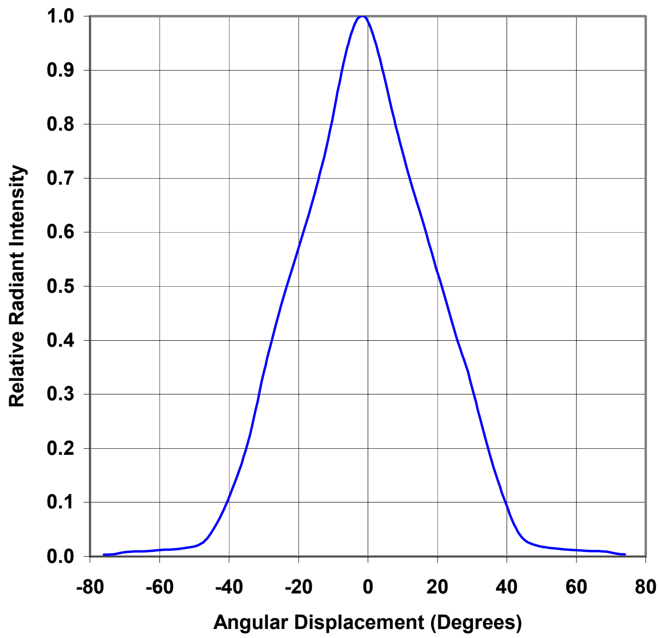
Optical Power vs Forward Current vs Temperature



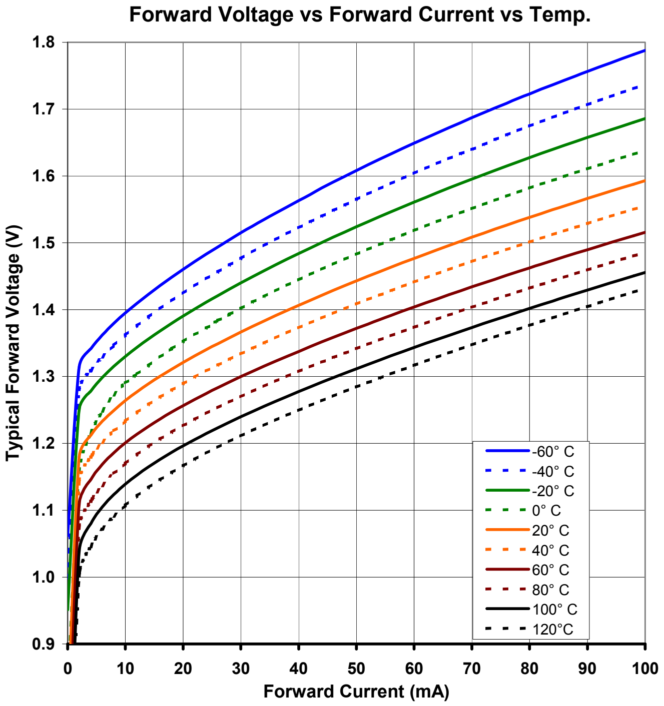
Distance vs Output Power vs Forward Current

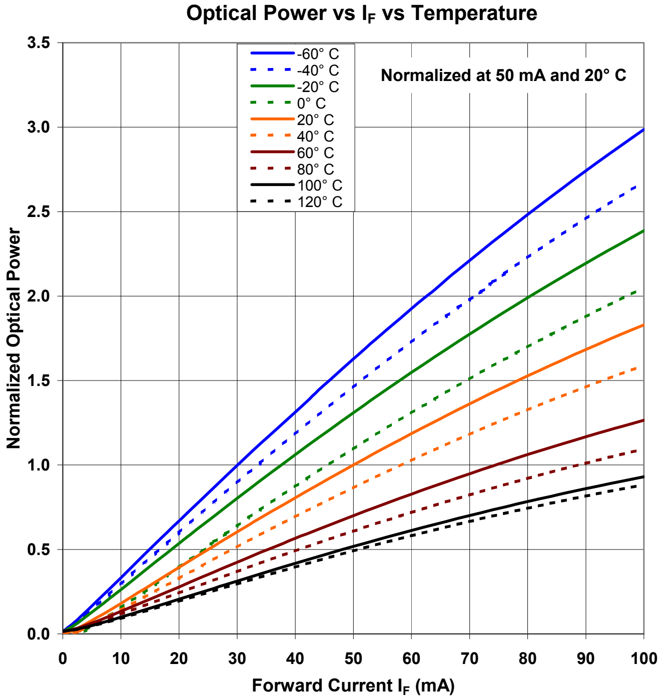


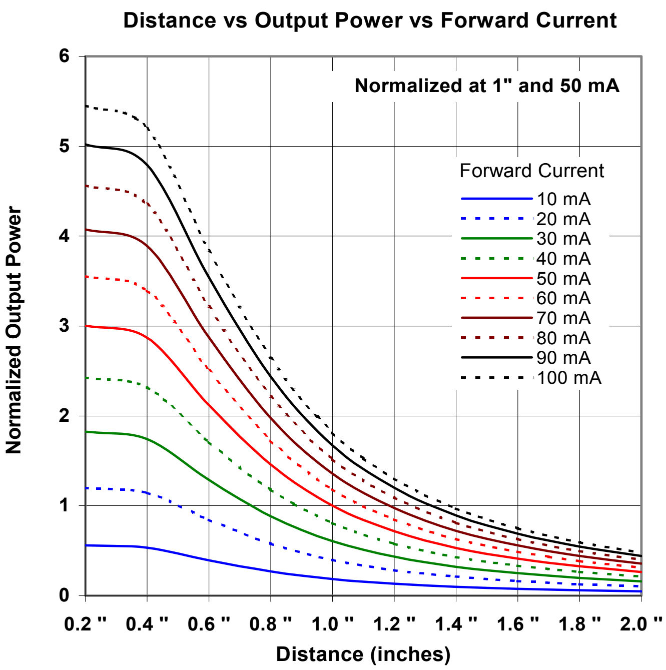
Relative Radiant Intensity vs. Angular Displacement

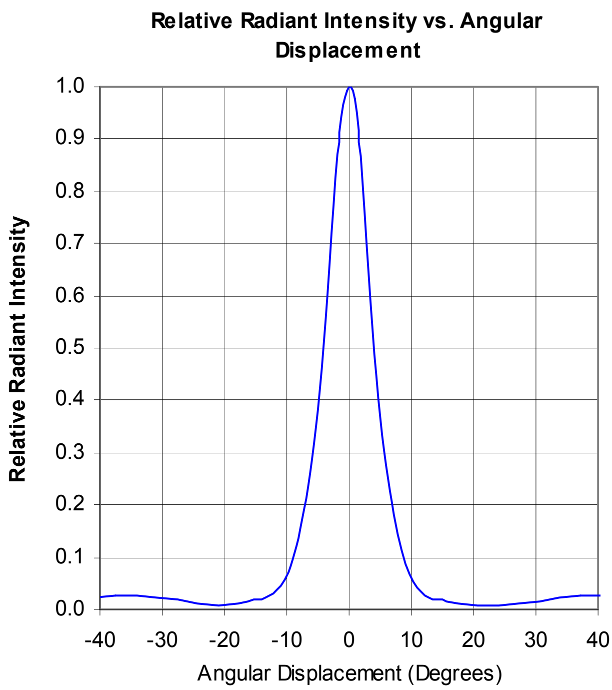


OP295, OP296, OP297, OP298, OP299

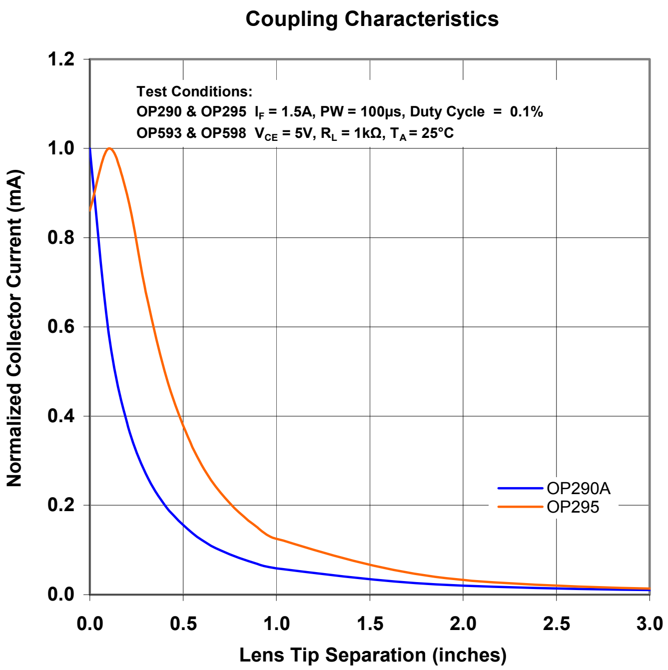








OP290A/OP593 and OP295/OP598 - Coupling Characteristics



OPTEK Technology Inc. — 1645 Wallace Drive, Carrollton, Texas 75006
Phone: (972) 323-2200 or (800) 341-4747 FAX: (972) 323-2396 sensors@optekinc.com www.optekinc.com

F

Contributions

All of the members have been involved in planning and implementing the prototype, as well as working with the thesis. The group has, however, had some separation on which area to prioritize.

The prototype

Fredrik Ehrndal and Pierré Reimertz have mainly focused on this part but all of the members have contributed and been involved with planning and decision making. Every member has been involved in constructing the hardware the prototypes.

The thesis

Filip Hesslund has been responsible for the layout and structure of the report, by learning LaTeX.

Below you will find what sections each person have been mainly focusing on in the thesis. Even though everyone has provided input on every section.

Joakim Bergström

1.0 Introduction (parts of)
1.1 Purpose and aim
3.2 Related Work
3.2.4 RollerMouse
3.2.6 Motion sensing input device
4.3 Design (Prototype)
4.4.1-4.4.7 Hardware
7 Conclusion

Proofreading

Fredrik Ehrndal

3.1 Ergonomics (intro)
3.1.1 RSI (Parts of)
3.2.1 Trackpad
3.2.2 Trackball
4.1 Requirements
4.5.4-4.5.8 Software
5.3, 5.4 Results
6.2.3, 6.2.4 Discussion
7 Conclusion Proofreading

Filip Hesslund

3.1.1 Repetitive strain injuries
3.2 Related work
4.4 Hardware
4.4.1 Arduino
4.6 Prototype iterations
6 Discussion
7 Conclusion
Proofreading.
Final design of the prototype.

Pierre Reimertz

1 Introduction
4.4.2 Phototransistor
4.4.4 Resistor
4.5.1 - 4.5.3 Software
5.1, 5.2, 5,4 Results
6.2.1, 6.2.2 Discussion
7 Conclusion
Proofreading

David Spånslätt

1. Introduction
1.1 purpose and aim (partally contributed)

1.2 delimitations
1.3 method
2 Idea intro
2.1 Design
2.2 interaction/movement patterns
3 Theory intro
3.1.2 ways to minimize the risks
3.1.3 keyboard usage guidelines
3.2.3 leapmotion
7. Conclusion