Kari Severinkangas

FTIR TOUCH PANEL FOR A MOBILE DEVICE

Architecture and prototype evaluation

FTIR TOUCH PANEL FOR A MOBILE DEVICE

Architecture and prototype evaluation

Kari Severinkangas
Master's thesis
Autumn 2013
Master Degree Program in Information Technology
Oulu University of Applied Sciences

ABSTRACT

Oulu University of Applied Sciences
Master Degree Program in Information Technology

Author: Kari Severinkangas
Title of Master's thesis: FTIR Touch Panel for a Mobile Device
Supervisor: Timo Vainio
Term and year of completion: Autumn 2013 Number of pages: 41+5 *appendices*

This work was done in a company named TactoTek, which is specialized in embedding electronic components into 3D molded plastic. The technology allows curved touch interfaces produced cost efficiently by integrating infrared transmitters and receivers into molded plastic. How to design an optical touch application that runs on a microcontroller of limited power and processing resources providing touch functionality that meets today's standards. The basic concepts of infrared detection and the frustrated total internal reflection (FTIR) phenomenon are gone through. The SW and HW architecture, signal processing and algorithms are described at high level.

A prototype device was developed with PC connectivity in order to read the runtime data to be displayed in the signal monitoring application and to feed the algorithm on computer. A simulator SW was developed to explore different kinds of methods for calculating touch points. The best algorithm candidate was integrated in a microcontroller providing digital interface for host device communication. The prototype's performance and accuracy was evaluated by using a robot tester and other manual measurement methods. In the end the prototype device was integrated in a cell phone with a rebuilt mechanics.

The developed touch application has an accurate multi touch functionality and allows input to be given with gloved hands. In addition, a touch size, shape and relative pressure are calculated. This study proves that this technology has potential and it can compete with the existing technologies providing a good performance, accuracy and a multi-touch functionality. The challenge in the technology is the activation sensitivity compared to the capacitive, which does not need a physical contact for a touch activation. On the other hand the FTIR panel provides a more natural and secure input method for applications where unintentional touches should be avoided. This could be advantageous in consumer applications where usually buttons and switches have been used, for example household and automotive applications. In order to achieve an optimal solution, more studies are needed in the material, hardware and signal processing.

Keywords: FTIR, touch display

TABLE OF CONTENTS

TERMS AND ABREVIATIONS

ADC	Analogue to Digital Converter.
ARM	Advanced RISC Machines, small microprocessor for embedded devices.
DAC	Digital to Analogue Converter.
DMA	Direct Memory Access. Allows transfers without CPU intervention.
EMI	Electromagnetic Interference.
FET	Field-Effect Transistor.
FPS	Frames Per Second.
FPU	Floating Point Unit.
FreeRTOS	Free Real Time Operating System. Free to use operating system.
FTIR	Frustrated Total Internal Reflection.
IR	Infrared light.
ITO	Indium Tin Oxide. Used e.g. in capacitive screens.
LED	Light Emitting Diode.
MCU	Micro Controller Unit.
MSPS	Mega Samples per Second.
PCB	Printed Circuit Board. Contains electrical components.
Photo receiver	Light sensitive component, Phototransistor or photodiode.
PWB	Printed Wiring Board. Contains no electrical components.
RAM	Random Access Memory. Memory that can be read and written.
ROM	Read Only Memory. Memory that can only be read.
SIP	System in Package. Integrated circuit including all the necessary components on a single chip.
SNR	Signal to Noise Ratio.
SOC	System on Chip. Number of integrated circuits put in a single package.
Touch controller	MCU and analogue components. Calculates touch positions.
Touch module	Combination of touch panel and touch controller.
Touch panel	Touch window, which is placed on top of display.
UI	User Interface.

PREFACE

In 2012 I had a great chance to start working at TactoTek to develop a touch technology based on a new manufacturing concept. Thanks to my boss, Antti Keränen CTO of TactoTek, for the support and letting me to public the Master’s thesis of the very interesting technology. Thanks to my colleagues in the prototype construction and HW related issues. Thanks to my family for giving me some quiet time at home while doing the documentation.

Oulu, Finland, September 2013

Kari Severinkangas

1 INTRODUCTION

1.1 Scope

TactoTek is a spin-off company from VTT (Finnish research center) that develops a novel FTIR (Frustrated Total Internal Reflection) touch panel technology. The patented manufacturing process (Pat. FI 121862B) makes the touch panel production low cost.

The study goes through the basic concepts of FTIR and IR (Infrared) detection and the most significant requirements focusing on a high level in the SW and HW architecture, signal processing and algorithms. One principle in the algorithm work was to do the research without using any other related studies in the same field.

At the end of the document a prototype device is evaluated to get an answer for the question, is it possible to design a low cost FTIR touch for a mobile device that meets today's standards? The study was done in a commercial company, for that reason all the details are not revealed, for example the algorithm part has been very short.

1.2 Technology

IR based touch displays have a very small role in the touch technology markets today. They are mainly used in large screens where they can provide a better performance per cost benefit and where the component size is not an issue. Tactotek's patented manufacturing process make the touch panel production low cost, efficient and allows the IR panel integration in small devices like cellphones and tablets. The components are placed in the edges of the touch panel, thus providing better visual performance. Other big advantage of the panel design is the cost scalability; doubling the touch area does not double the component count, the scaling goes merely as the screen perimeter length.

The touch detection in the Tactotek's panel is based on a physical phenomenon called FTIR (Zhu et al. 1986, 601). Infrared light is sent into a waveguide, plastic or glass. The light traverses the waveguide without any significant attenuation because of total internal reflection, as in optical fibers. When the surface is touched, total internal reflection is disrupted (i.e. light escapes from the waveguide) and the intensity of the received light is reduced, which can be measured (see chapter 2.1).

TactoTek's manufacturing process (Pat. FI 121862B) is very simple and cost-efficient, LEDs, photodiodes, controls and wiring are put on a foil substrate and the screen cover (waveguide) is molded on top of foil. The process gives a significant cost advantage, both in terms of touch panel bill of materials and factory investment.

1.3 Touch technologies and markets

There are many different types of touch devices in the market today, e.g. smartphones, tablets, MP3 players, car navigation devices. The growth of the touch screen modules has been very strong and the forecast shows still strong growth (figure 1). The demand is growing for thinner, light weight and lower cost touch panels and devices.

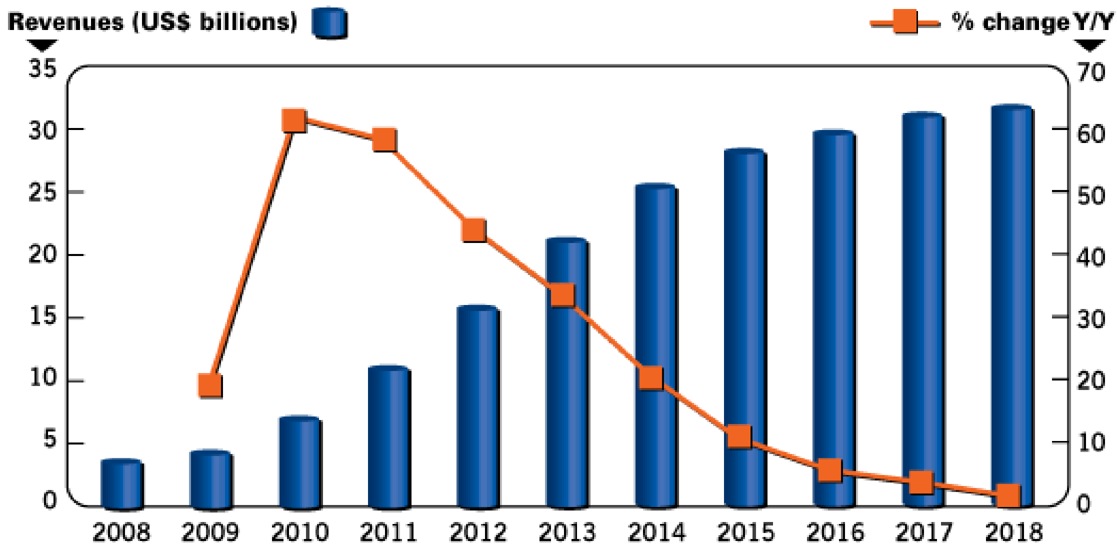


FIGURE 1. Touch screen module revenue forecast (DisplaySearch 2012, date of retrieval 5.6.2013)

The projected capacitive technology is growing fastest. Figure 2 shows the forecast from IDTechEx for market size in 2012. Robustness, performance, multi touch and self-calibration have been the key factors for the success. The capacitive technology has become almost a paradigm in user interfaces, which makes it difficult to challenge. One may argue whether the usability is good, for some applications capacitive displays are too sensitive and they lack natural interaction with feed-back.

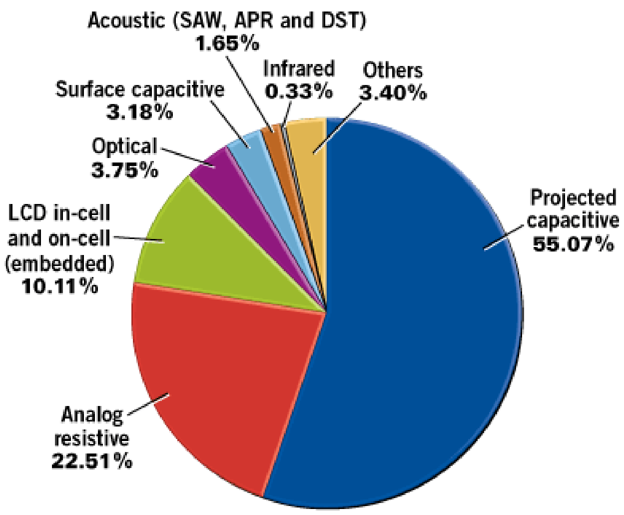


FIGURE 2. Touch market forecast by technology in 2012 (IDTechEx 2012, date of retrieval 8.9.2013)

There are many different methods to measure touch and each method has its benefits and drawbacks. The different touch technologies are compared in table 1. Interacting with the device should be smooth and natural allowing the user to give input in different situations by using a stylus, finger or glove. Touch resolution is very high in small capacitive touch devices, accuracy can be around 0.1 mm in the middle of the screen. For finger input this is more than enough, there are not so many applications that actually need so high resolution. Multi touch capability has become a must, at least gestures like pinch and zoom should be possible. Many small devices (3-4 inch) support up to ten fingers, but no real use cases can be imagined for such many fingers.

The optical transmissivity is an important feature, different layers and air gaps between touch surface and display weaken light transmission and cause reflections. Scalability and cost are important factors for a touch module manufacturer. In which ratio the touch module costs increase is related to the screen size, i.e. the cost per inch. In bigger screens, above seven inch, there are opportunities for other technologies as the capacitive solutions do not scale very cost effectively. With some technologies it is challenging to scale down due to physical component sizes.

The resistive technology is still used in low cost devices. The resistive touch panel is made of two ITO layers with spacers between them. The top ITO is laminated to the scratch resistant flexible plastic and the bottom ITO is on the rigid base (e.g. glass). Touch is detected when ITO layers enter into contact. By applying power to one layer, the touch position can be measured from the other layer by measuring the voltage level. The resistive touch allows input with any object, but in most devices only single touch is permitted. Also some force is needed to active touch, which makes swipe gestures clumsier.

There are quite many different types of capacitive sensing methods. Also, the sensors can be built with different stack-ups, they can be separate modules having air gap between a display and a sensor module, laminated on a display or integrated in a display (on and in-cell). All structures require a conducting layer (e.g. ITO) on the visible display area. There are two types of capacitive sensing methods, surface and projected. In the surface capacitive technology the screen glass has a conductive layer. By applying voltage to all four corners the electric field is formed. A finger touch causes a voltage drop in each corner allowing a touch position calculation.

The projected capacitive screens can be divided in mutual and self-capacitive. In the self-capacitive touchscreen, the touch panel is made up of a grid of individual sensors. Each sensor line is measured individually; the increased capacitance in a line indicates finger presence. This sensing method allows only single touch and dual touch gestures. In the mutual touch the sensors form a grid where rows and columns form individual intersections. Current is driven to TX lines and RX lines are read in turn, the mutual capacitance between row and column is reduced when finger is in presence. The mutual capacitive sensing allows a multi touch functionality. (Global Display Engineering 2012, date of retrieval 8.9.2013.)

The draw-back of capacitive technologies is that the input method is limited to the finger, as the human body capacitance is needed. The use of thick gloves or no conductive objects is not possible. Some devices can detect fingers through thin gloves, but the draw-back is a too sensitive interaction without gloves. Other issue is the cost scalability when the display size is increased (except surface capacitive).

In big screens the optical technologies have been used for some time, as they have a good cost per size ratio and performance. Having sensors only in the edges has a cost and transmissivity benefit. By increasing height and width to double, the component cost is only doubled even though the display area is quadrupled, glass or especially plastic window having a minor influence on the total cost.

In the IR technology area there are two types of methods for sensing touch, free air and an FTIR based. In the free air IR light is transmitted over a display. When finger approximates a display, it breaks the light path causing a strong attenuation in the receiver. The free air IR touch is immune to dirt and it works with any objects. The draw-back is that it requires some space in the edges of the display, not allowing making sleek designs. Neonode is one of the companies in the market specialized in this method. The FTIR touch needs a separate window where the IR light is transmitted. There must be an air gap between the touch window and the display. There is one company, named FlatFrog, that has been developing the FTIR based touch technology, but only for big screens (>23 inch). There are also optical systems where the camera is on the other side of the screen, obviously these systems are used e.g. in the interactive tables but not on displays.

TABLE 1. Different touch screen technologies compared (Global Display Engineering 2012, date of retrieval 8.9.2013).

	Resistive	Projected Capacitive	Surface capacitive	Infrared (free air)	Acoustic (SAW)
Transmissivity	<82%	>92%	>92%	100%	>92%
Activation	Moderate pressure	Low pressure	Low pressure	Zero pressure	Low pressure
Durability	fair	excellent	excellent	excellent	Good
Cost	low	medium	medium	high	medium
Typical Display size	Small <19"	Small <19"	Small <19"	Very large >5.7"	Medium 5.7-30"
Multi-Touch?	not typically	yes	no	yes	Not typically
Input method	finger or stylus	finger or conductive stylus	finger or conductive stylus	finger, stylus, etc.	finger, soft tip pen
Challenges	durability, needs force full touch	Glove use	Glove use	sunlight, contaminati on, high bezel	rain, dust, contaminati on

1.4 Stakeholders and customer needs

It is critical to identify the needs of the stakeholders of the system (see table 2). The most important stakeholders are the end users and the device makers. The touch module must meet today’s user expectations in response time, sensitivity and usability. Device makers need easy to integrate, low power and low cost solutions.

The touch module manufacturer assembles the components (transmitters, receivers and controller IC) on a foil and molds the waveguide on top. The IC manufacturer makes the customized IC, which is a MCU with the necessary peripherals and analog components.

TABLE 2. Stakeholders and needs

Stakeholder	Needs
Device manufacturer	Easy to integrate, low power, low cost, durable
End user	Responsiveness, sensitivity, usability, durability
IC manufacturer	Simple and standard HW design
Touch module manufacturer	Easy manufacturing process, high yield

1.5 Critical requirements

In the beginning of the project the most important requirements were identified. We call these requirements “sacred” as they are crucial to the project.

- Low price. Competitive with the resistive and low cost capacitive technologies.
- Competitive performance compared to low cost capacitive touch technologies: frame rate, accuracy, sensitivity and linearity.
- Multi touch capability.
- Immunity to EMI
- Competitive power consumption (less than 20mW).

1.6 Study objectives

This study describes my engineering work, which includes the SW/HW architecture, touch detection algorithm including MCU integration and the basic concepts of FTIR. The automated algorithm tester and the prototype HW were implemented by my colleagues. A prototype was constructed during the work to get a better understanding of the challenges and possibilities of the technology. The first version of the panel was made of phototransistors. Phototransistor was chosen, because from an HW design point of view it is an easier component to integrate compared to a photodiode. Photodiode and phototransistor are described in chapters 2.3 and 2.4.

The algorithm's accuracy and performance is evaluated in a generic microcontroller development kit. The SW architecture of the microcontroller, scheduling, signal processing and calibration are described.

2 FTIR AND HW OPERATION

2.1 Total internal reflection

The FTIR phenomenon has been known at least from the times of Isaac Newton (Zhu et al. 1986, 601). When the angle of light reaches the critical value, it suffers from the total internal reflection (figure 3). The critical angle depends on the refractive indices of n1 and n2 and can be calculated by using the formula $\sin \Theta = \frac{n2}{n1}$ $(n1 > n2)$ (figure 3). The formula is calculated from Snell's law (Benson 1995, 715). For example polycarbonate has a refractive index of 1.58 and air approximately 1, applying these to the formula gives 39.2 degrees. A finger or any object that has a greater index than 1 will change the critical angle to a bigger value thus frustrating the light.

When total reflection occurs interesting waves can be observed in the second medium (n2 in figure 3). These waves are called evanescent waves. They penetrate in the other medium approximately in the range of wave length or less of transmitted light depending on the reflection angle and medium indexes. In theory this means that reflected light can be disturbed in the range of evanescent waves. (Zhu et al. 1986, 604-606.)

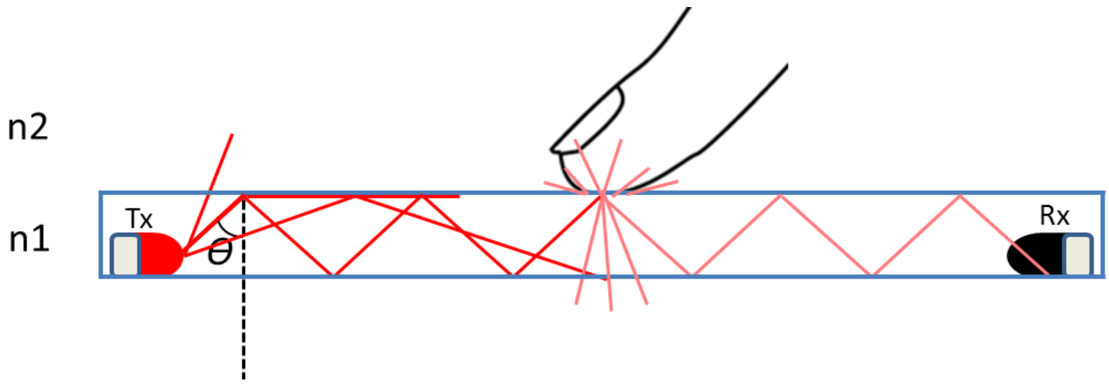


FIGURE 3. Principle of FTIR: Finger attenuates the signal, which is detected in receiver.

Components are inside the plastic in the FTIR panel, thereby if the components have plastic lenses (most in the market) they do not have much effect on the intensity spectrum. This means that LED has an equal intensity distribution if looking from top or from the side, radiation follows Lambert's cosine law (Cohen 2013, date of retrieval 6.11.2013), which is illustrated in figure 4. The same applies to the detector, the detection sensitivity is the highest for light perpendicular to the detector chip surface and it decreases to zero at 90 degrees (light parallel to the chip surface). Naturally, the material of the FTIR panel and the photo component package (including lens) has some influence on the overall performance.

.

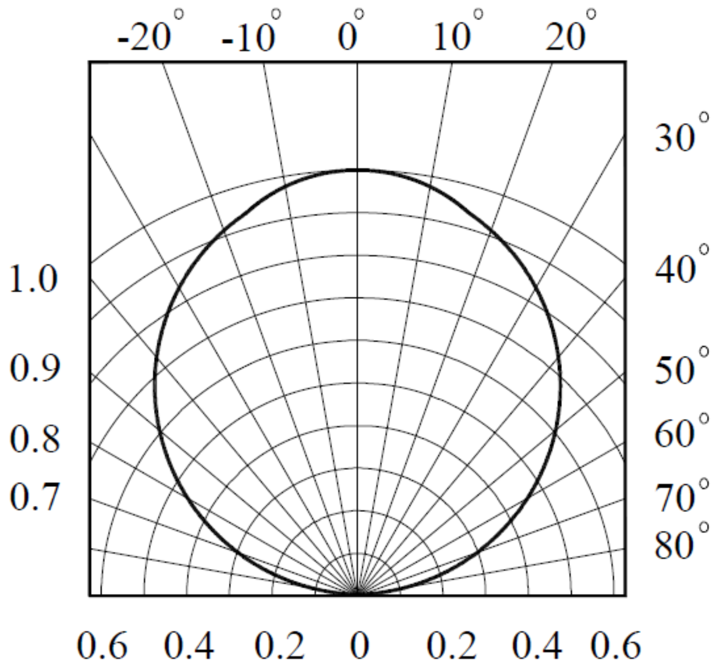


FIGURE 4. LED's relative radiant intensity vs. angular displacement (Jaine 2011, 5).

2.2 Touch panel

In the touch panel light transmitters and receivers are placed in the edges of the touch panel forming a grid, which allows the touch detection. Each transmitter is flashed with a short pulse and response is measured from several receivers. This scanning method reduces the power consumption and enables a multi touch detection and high resolution.

The touch panel contains LEDs and photo receivers on PWB (Printed Wiring Board), which may be flexible (figure 5). The trend in the mobile devices is that the "dead band" on the display frame is very small. Display area, including touch, should cover the whole front side of the device. This sets a challenging requirement for the flex design: components and wirings must be tightened-up by using more wiring layers or bending flex in the phone front cover. By using an innovative 3D modeling they can be embedded in the mechanics. Decorations, which must not disturb IR light, can be used to hide components. The curved design support is one of the benefits of the FTIR panel, thus it gives quite free hands for device designers.

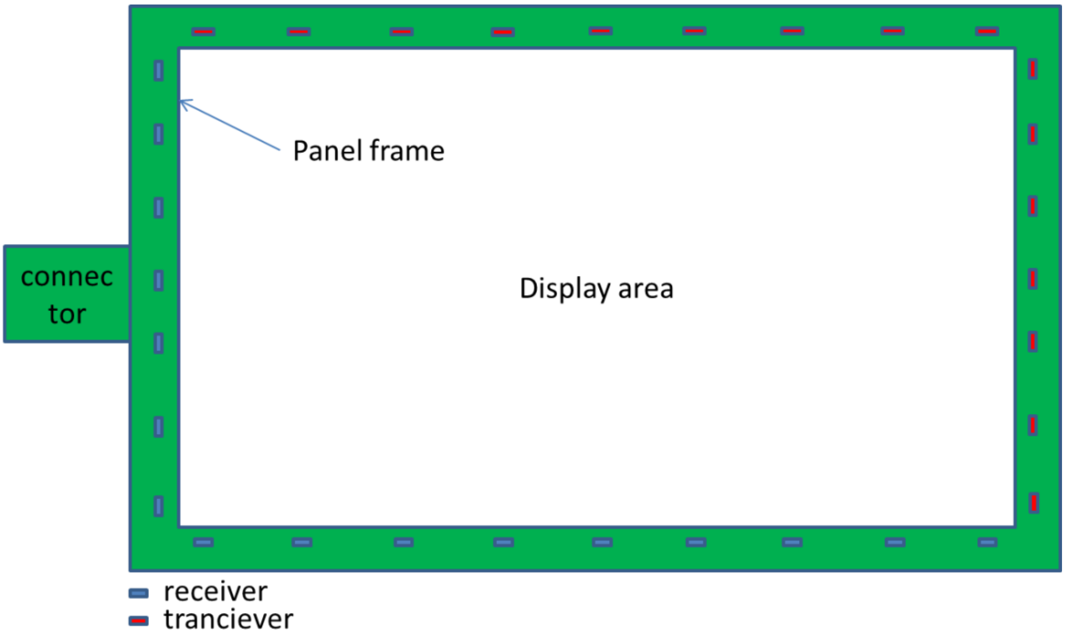


FIGURE 5. FTIR touch panel

In order to achieve a narrow dead band and a low cost design, the component count and signal routings to MCU must be minimized. It is very important to achieve a good SNR between the panel and MCU and keep EMI issues minimal. LEDs and photo receivers must be small and power efficient and they should work in the narrow IR light band. The latter requirement is important because we only want to measure the IR LED pulses in the system not the ambient light.

2.3 Phototransistor

Phototransistors are based on regular transistors with the difference in the base part, which is exposed to the light. Light on base causes a current between the collector and the emitter. A common collector circuit is presented in figure 6.

The resistor size defines the output voltage. The bigger resistor value increases the time constant and raises the voltage level. A better resolution from the ADC converter can be achieved by raising the voltage level, but on the other side the signal rise times grow. Also, it must be made sure that the collector emitter voltage (V_{CE}), and the collector current (I_C) are at the sufficient level. In the end a good compromise needs to be found

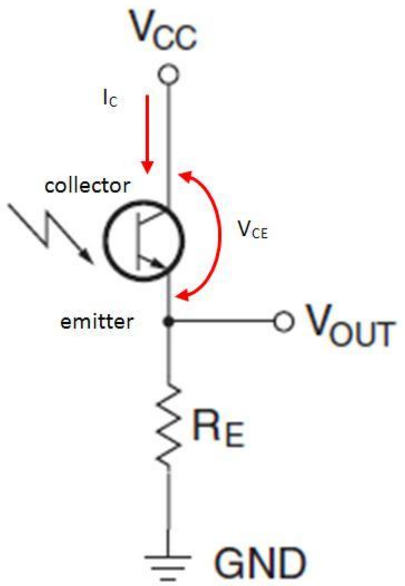


FIGURE 6. Phototransistor measurement circuit

Theoretical rise and fall times for a phototransistor are around 15 µs. In practise it is challenging to achieve this speed and maintain a good sensitivity.

2.4 Photodiode

Photodiodes are based on regular light emitting diodes with the exception that they have an exposure to the light sensitive part. Photodiodes can be designed to work in different modes, photovoltaic, positive and negative biased circuits. Figure 7 shows a positive biased example.

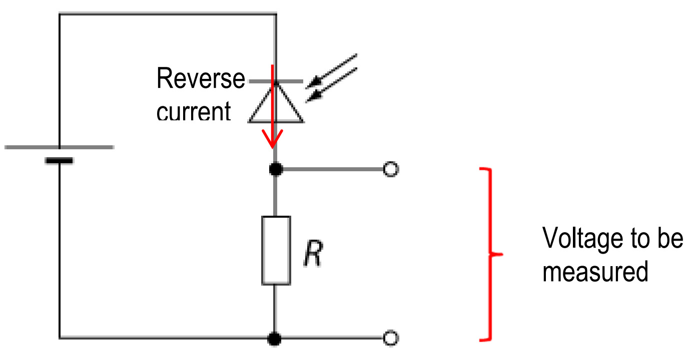


FIGURE 7. Photodiode positive bias circuit

The measurement with photodiodes is similar to phototransistors, except that the current is much smaller, thus the voltage to be measured over resistor is much smaller. This places more challenges for signal routing from panel to MCU in order to keep SNR at the acceptable level. Photodiodes are very fast typically 6 ηS rise/fall time, but in practise this is difficult to achieve.

2.5 LED pulsing and photo receiver reading

Each photo receiver has a set of LEDs that it can detect. Because of the Lambert’s cosine law, it is not possible to have all theoretical channel combinations. Figure 8 illustrates what LEDs a photo receiver can see in the example panel layout. The drawn lines are can be called channels (or scanning lines).

Each channel carries information about light intensity. For the panel in figure 8 there is at the maximum 256 (16*16) channels. In practice the channel count is less due to Lambert’s cosine law. MCU pulses LEDs and reads light intensity from receivers by using an Analog to Digital converter (ADC). Each time one LED is pulsed and one or more receivers are read.

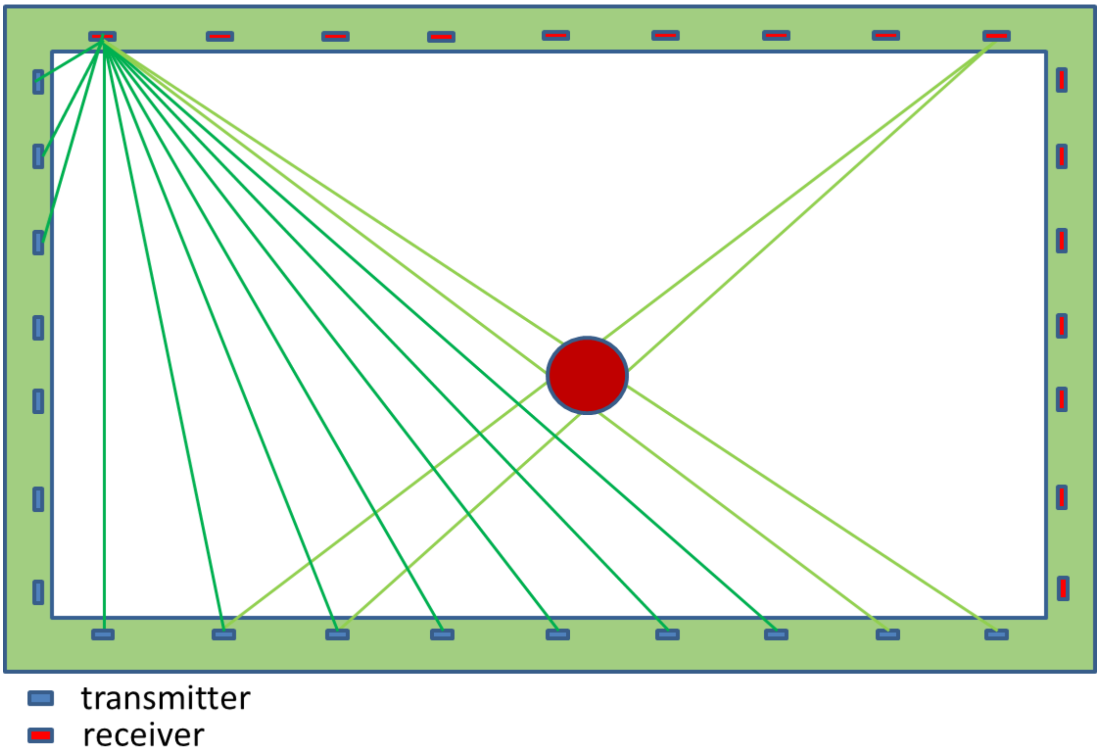


FIGURE 8. Example of panel structure with components.

Figure 9 shows how the analog signal is read. The blue toggling line tells when LED is switched on. In this example case a phototransistor takes about 20 µs to stabilize. MCU switches LEDs by using an HW timer. Every 20 µs an analog signal is read and LED is pulsed (on/off). The receiver also reads the ambient light level, as it must be subtracted from the LED pulse value in order to avoid shadows and other interferences to disturb the touch detection. When all channels have been read, a data frame is ready for a further signal and algorithm processing.

The receiver's speed requirement is defined by the following factors: target frame rate, LED power consumption, total channel count, EMI issues and the ADC conversion time. When designing a new panel, the screen size and the needed resolution must be known. These requirements define how many components are needed. The target frame rate is another requirement that defines frame's scanning time, which in turn sets a requirement for receiver's rise/fall time. Another requirement is power consumption; a bigger rise/fall time means more LED burning.

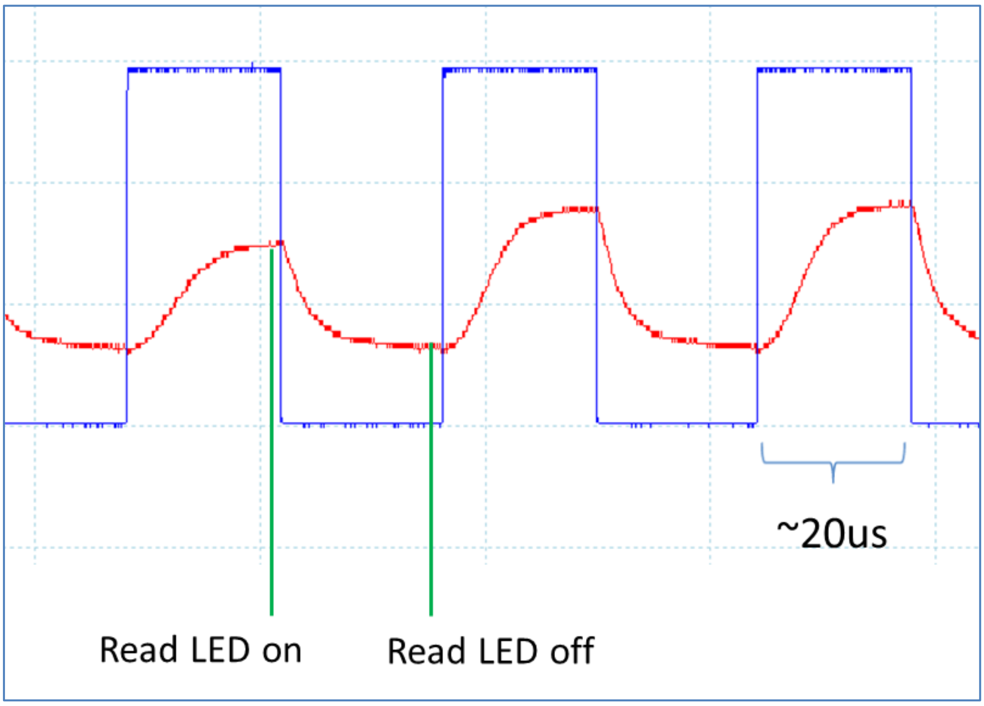


FIGURE 9. LED pulsing (in blue) and analogue signal response (in red)

In order to detect touch, there must be a reference level where to compare the current channel value (figure 10). When a device is booted, it is calibrated and the reference signal levels are stored to the volatile memory. After the calibration, the signal levels read from ADC are compared to the calibration levels. When a threshold level is passed, a touch is detected. Because of dirt and water (or liquids) a slow calibration task must be run continuously on the background.

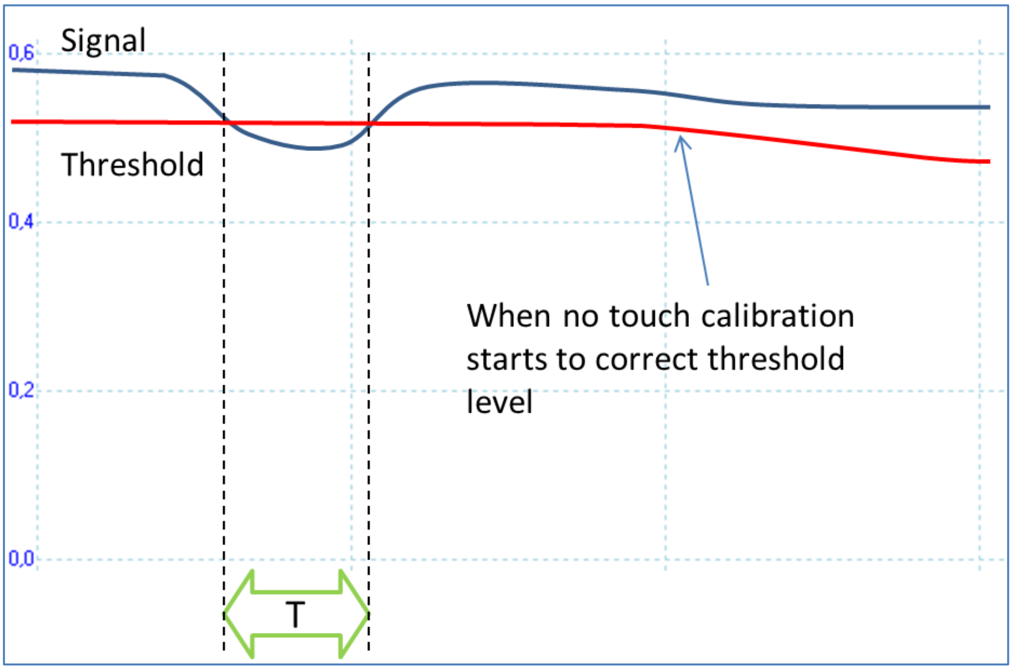


FIGURE 10. Touch detection and calibration

3 ARCHITECTURE

3.1 Functional architecture

Figure 11 shows the main functions of the FTIR touch controller. The touch controller provides an interface for host communication. The communication protocol must be well specified and flexible and not be tied to the underlying SW and HW implementation. The touch controller notifies host about new touch events, which should be read immediately. The touch controller provides a command interface for configuring touch detection related parameters, different power modes and self-test execution. The touch controller has a self-adjusting power mode algorithm, which changes the scanning frequency and enables/disables analogue and digital peripherals.

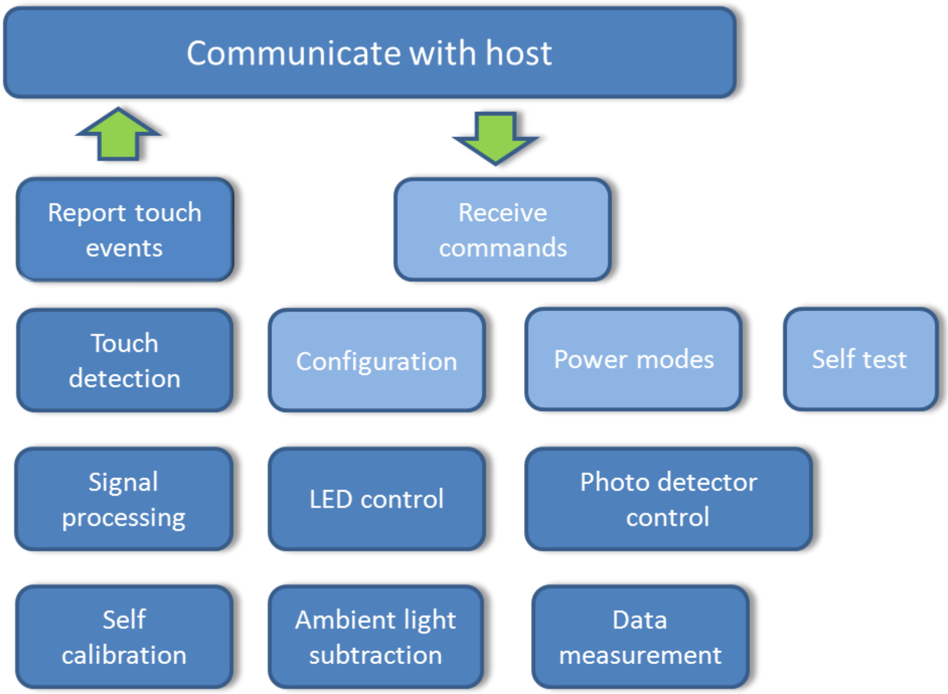


FIGURE 11. Functional architecture blocks

In data measurement the IR light intensity is measured and converted to digital. An ambient light subtraction must be done in order to eliminate all unwanted light sources. MCU drives LEDs and photo detectors according to the current power state. In signal processing an analogue and a digital signal is filtered to remove all undesired noises in the system. The signal is then normalized and pre-processed for the touch algorithm (touch detection). Self-calibration is an important function as it adjusts the sensitivity in runtime to the optimal level eliminating effects of dirt or water on the touch panel.

The next chapter describes the physical architecture. The SW architecture is described in chapter 4.

3.2 Physical Architecture

The target is to find an optimal physical architecture where the performance, cost and size requirements are met. Figure 12 shows how the functional blocks are partitioned in the prototype device. The HW module (i.e. touch controller) provides a digital interface (I2C), which is used for communication. The interrupt line tells the host device when it should read new touch coordinates. The signal from the photo detectors (from touch panel) is amplified and low-pass filtered in the analogue part. It also contains a circuit for ambient light elimination. MCU drives LED and photo detector matrixes and triggers the ADC conversion, which converts the signal into a digital format. In the final product MCU and peripherals are integrated into SOC (system on chip).

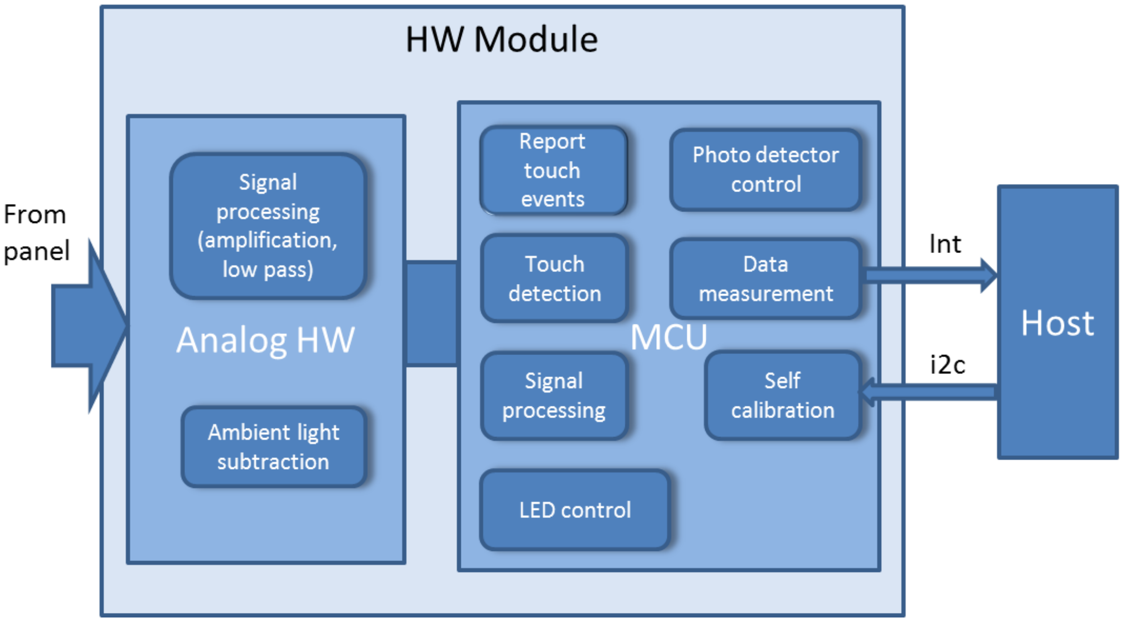


FIGURE 12. Functional partitioning

3.3 Verification and architecture choices

Architecture is validated by prototyping and using iterative SW development methods. Also some quality methods can be used, for example back tracing from physical HW and SW components back to requirements. Each component in the system should have a purpose and fulfil a requirement. Table 3 shows high level relationships between components and sacred requirements. During the project the status of the requirements is followed and actions are taken until each requirement has been fulfilled.

TABLE 3. Requirements and related components

	Sensitivity	EMI immunity	Power consumption	Multi touch	Frame rate
Touch panel	x	x	x	x	x
Analogue HW	x	x	x		x
MCU	x		x	x	x
Touch detection	x		x	x	x
PCB		x			x
Signal processing	x	x	x		x

There are many options for HW selection. The panel component layout can be constructed in many ways, the amount of photo detectors and transmitters can vary. For the IR photo detection two types of components can be chosen, a photodiode or a phototransistor (see next chapter).

There are many companies selling standard microcontrollers for different applications. The FTIR application is a special case, thus it can be difficult to find a feasible MCU from the market. Using the standard MCU would give a fast time to market benefit, but cost would not be the lowest. The most cost efficient method is to make a custom SOC (system on chip), but the initial development cost is high, which means that the manufacturing volume must be in millions. A compromised solution would be to use SIP (system in package), where a number of integrated circuits are combined in a single package, containing MCU, memory, peripherals and passive components. The selected HW configuration should be scalable from small displays to larger ones, for example 3-7 inch.

3.3.1 Photo receiver choices

Frame rate and power consumption depend directly on the rise/fall time of the photo component. When thinking of power consumption, it is quite clear that a photodiode is a better option (table 4). A faster rise/fall time also means less LED burning. A disadvantage of a photodiode is that its signal must be greatly amplified. Routing a very low amplitude signal is very challenging, especially in mobile products that have a lot of interference noise. Another challenge with a photodiode is the matrix circuit. Phototransistors can be driven by using a HW matrix made of FETs. This reduces signal tracks to the panel, e.g. for 16 phototransistors only 8 signal tracks are needed (4x4). In a case of a photodiode the driving circuit could be placed near the component like in camera sensors, but this means more electronics in the panel, which in turn increases the complexity and risk in the manufacturing. Another option is to use less photodiodes without any HW matrix.

TABLE 4. Photodiode compared to phototransistor

Photo receiver	Photodiode	Phototransistor
Rise/fall time	6nS	15uS
Current consumption	In uA	In mA
Price	2 cents (Euro)	2 cents (Euro)

3.3.2 MCU selection

The MCU selection is always difficult in the beginning of the project. For example, how much processing power and memory is needed. Prototyping and iterative development is a good approach when there is too much uncertainty. Some rough estimates were done in the beginning of the project.

- Equal or lower performance than 32-bit ARM cortex M3 processor (68 MHz), no FPU unit.

- Fast ADC converting time providing 1-2 Msps performance (12-16 bit).
- 16 kb of RAM
- 64 kb of Flash

Today a 32-bit processor is a good alternative as its price has gone down. It is power efficient and has a good code density. Cortex M3/M4 uses Harvard architecture, which has own busses for code execution, data and peripheral control, unlike Cortex M0/M1 (appendix 2). RAM memory is more expensive than flash memory, which means that large data tables must be kept in flash memory. This has some performance penalty, especially if no memory accelerator is available. In the end the ADC performance requirement will depend on the achieved photo detector speed and the overall frame rate requirement of the system.

4 SW ARCHITECTURE

4.1 Overview

The HW requirements dictate very much the SW requirements. The SW binary size, the dynamic memory and the available processing power are limited. Usually FTIR based systems have been built for devices that have a lot of processing power and an unlimited electric source. The algorithm for low end MCU with a battery supply must be simple and efficient.

The SW architecture (figure 13) consists of two parts, MCU and a PC. The algorithm is developed on a PC using simulation or raw measurement data. Two versions of algorithm are maintained, Python and C version. The Python language is very popular these days and provides various different kinds of modules for fast development. The C algorithm can be tested on a PC with the same tools as Python. This saves a lot of testing work on MCU.

The Python FTIR component reads the ADC data from MCU through USB. It does quite the same things as C FTIR in MCU. The test tool reads the measurement data from the Python FTIR component and visualizes it in different ways or feeds the data to the algorithm and visualizes the detected touch positions in UI. The algo tester is a test automation tool, which uses the simulator for generating simulated data. It can iterate through all possible input combinations with different sizes of finger, different algorithm parameters and panel configurations. It can also read the raw data from the database, which makes the testing more realistic.

C FTIR has the following tasks to perform:

- Drive the analog front end
- Schedule and switch LED and photo receiver matrixes.
- Perform dynamic LED current adjustment
- Converse analogue photo receiver signals to digital.
- Perform digital signal filtering and run-time calibration.
- Send data over I2C when master requesting or receiving commands from master.

The C FTIR component fills the frame data buffer and feeds the pre-processed data to the algorithm. The C FTIR component also interfaces with the MCU and drives the analogue HW.

The HW driver represents the low level SW stack, which interfaces with MCU. The C Algo (algorithm) component does the actual touch position calculation.

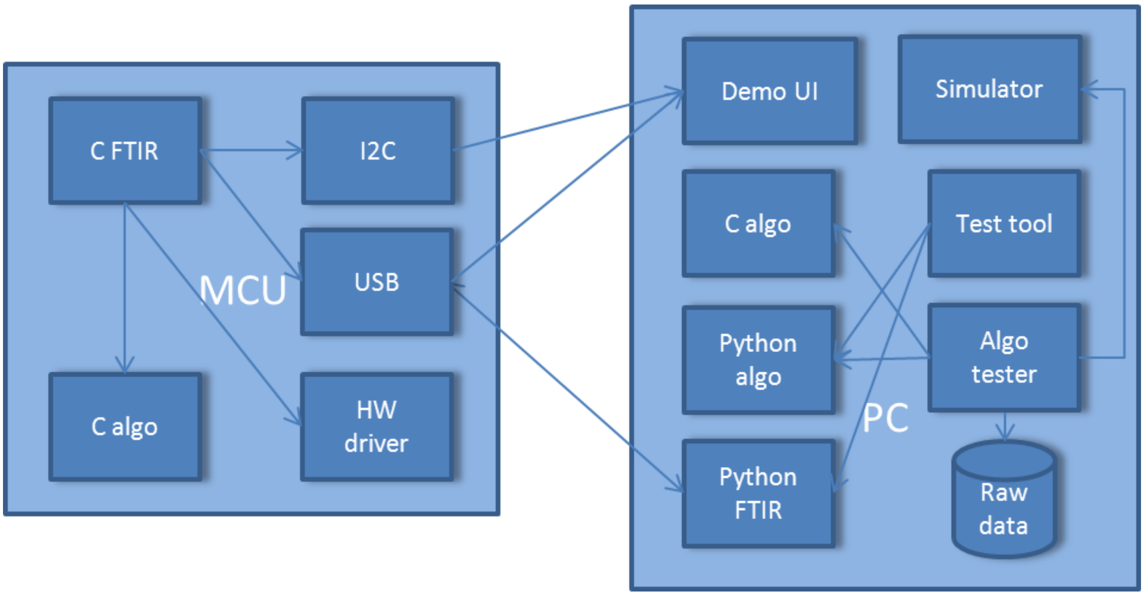


FIGURE 13. SW architecture

The touch algorithm supports any size of touch screens. When the resolution or panel size (and component count) is changed, the algorithm is unchanged. Obviously when increasing the touch screen size, the memory requirement increases, too. The panel and algorithm configuration is defined in files. The Python test tool reads the configuration files and generates arrays in memory when started. The test tool includes also a C code generator, which reads the files and generates all the necessary constant arrays for the C algorithm.

4.2 Testing

The architecture supports testing at different levels. The C algorithm is unit- and module tested on a PC. The Algo tester does the module testing. It takes the input from the simulator or from the database containing the unfiltered measurement data. The measurement data can be read to the database during a robot testing.

Figure 14 illustrates the testing set-up. Test SW is run on a PC, which is connected to the test robot controller and the touch controller. The device under test (DUT) is the touch controller and the touch panel. On the PC the Python tester SW reads the test input from the database and commands the robot application. When the robot has placed the test head to the desired position the test SW reads the touch controller (n touch events) and stores the output to the database. Another option is to store the raw measurement data to the database for later use with the algo tester.

Test robot should have various test heads simulating different kinds of human fingers having options for shape, size and material. This is one challenging part in the testing. The raw measurement data should be collected from various people to find a range of finger types that the FTIR system must be able to detect. For example detecting dry fingers is challenging and requires a very sensitive and noiseless system.

The robot tester allows an automated testing for different panel sizes. It also enables an easy algorithm development by storing raw measurement data to the database.

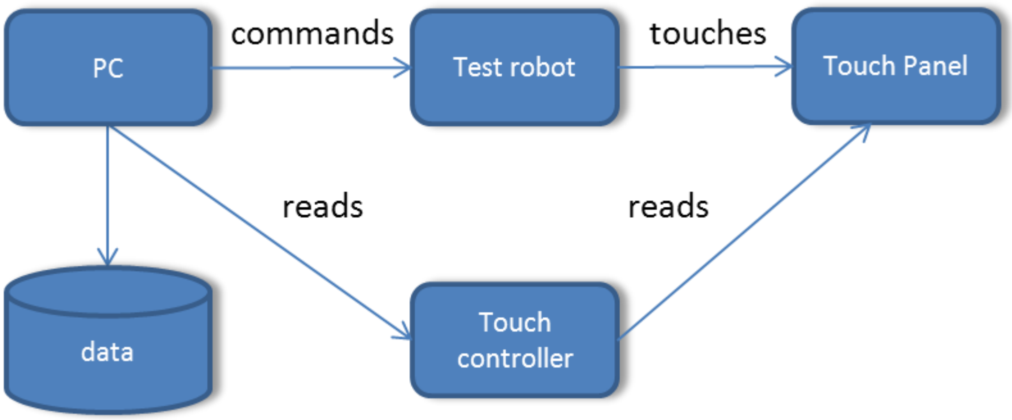


FIGURE 14. Testing set-up

4.3 Scheduling

In the beginning of the project it was unclear how the final scheduling will be done. To make things easier to start, the FreeRTOS scheduler was chosen. The FreeRTOS memory requirements are very low starting from 500 bytes.

There are several tasks to perform by MCU. DMA should be used whenever possible as it releases CPU time and eases scheduling. Measurement data is read constantly from the ADC block and calculated touch positions are written to the I2C block. Both of these HW blocks should use DMA for data movement.

Figure 15 illustrates at high level how scheduling works with FreeRTOS. The ARM processor has an interrupt vector table from where it branches to the interrupt functions in the FTIR code.

There are three relevant interrupts: SysTick, timer and I2C. The FreeRTOS scheduler uses ARM SysTick (system timer) interrupt for scheduling an interval, the default value of which is 1ms. The timer interrupt has the highest priority, because it must pulse LEDs with no latencies. The I2C interrupt has the second highest priority. SysTick has the lowest priority as it wakes up the scheduler, which triggers the relatively long time algorithm task.

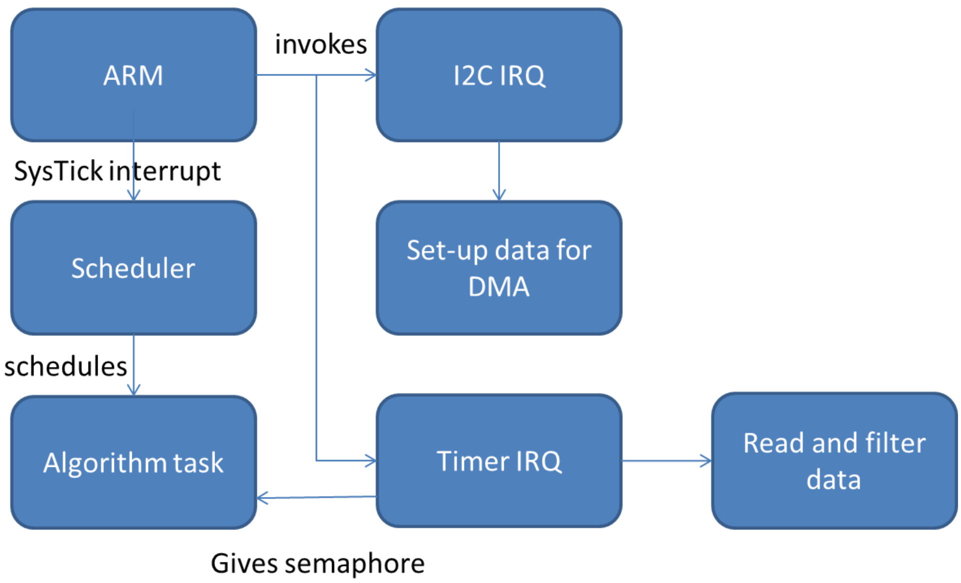


FIGURE 15. Scheduling by FreeRTOS

When there is no other interrupts in queue, the SysTick interrupt triggers the scheduler and the algorithm task gets started or resumed. When I2C or Timer interrupt happens, the algorithm task is stopped and the registers are put on stack. The algorithm task is enabled (semaphore is given) by timer interrupt after all channels have been scanned, i.e. the frame buffer is ready. In FreeRTOS the binary semaphore can be used for task synchronization.

There is also an option to leave the scheduler out by enabling a flag. In that case the Timer IRQ function calls the algorithm function after a frame buffer is ready. This means that the algorithm function is executed until the end and no other interrupt can stop the function. Also, it means that during LED pulsing (timer on) processor time is not used.

A more detailed scheduling diagram can be found in Appendix 1. The SW and HW were implemented without an external interrupt line connected to the host. The host system polls the touch data at about 100 Hz. This means that the touch data buffer must be protected from being modified and transmitted at the same time. When using an external interrupt, the host will usually read the fresh data right after the interrupt before the next data is available.

4.4 Algorithm

4.4.1 Choosing algorithm

The algorithm can be qualified by accuracy, speed and scalability. The scalability has various parameters. A scalable algorithm continues to perform well when increasing a touch area (panel size). Also, it should be easy to change the touch panel without making any significant SW modifications. In the system the speed depends mainly on the frame rate, digital/analogue filters and algorithm. The algorithm must calculate the results from the input with as few CPU cycles as possible. The algorithm should be simple and efficient to be run on a low cost microcontroller. Speed and accuracy are analysed in chapters 5.3 and 5.4.

In general, implementing SW is balancing between a CPU use and a memory. By increasing the memory, the CPU load can be reduced. One approach is to calculate everything on runtime. Another approach is to calculate the necessary data beforehand and store it to ROM when building the executable.

4.4.2 Algorithm interface

The algorithm reports the following data:

- Type of touch event (down, up, move)
- X and y coordinates (center point of touch)
- Average width and height of touch
- Pressure of touch.

Usually a host device can configure a touch controller to report the wanted data. Coordinates and an event type are typical values to report. In the FTIR application the pressure tells directly the average level of attenuation in the channels under touch. The algorithm does not calculate the exact touch shape, as it would require more processing, but average values of width and height. After all touch areas have been calculated, the touch data is put in a simple byte buffer to be send over a communication channel when requested by the host. The more detailed picture of scheduling can be found in appendix 1.

5 RESULTS

5.1 Prototype HW and SW

The prototype HW used in the project is a sandwich solution containing a MCU part on one circuit board and an analogue part on another board (appendix 3). The MCU board is a high performance STM32F4-Discovery kit. The MCU kit has the following key features:

- ARM 32-bit Cortex™-M4 CPU with FPU (Floating Point UNIT)
- frequency up to 168 MHz (1.25 DMIPS/MHz)
- 1 Megabyte of Flash memory, Up to 196 Kbytes of SRAM
- 3×12-bit, 2.4 MSPS ADC: up to 24 channels
- 2×12-bit DAC, 3 × I^2C, 16-stream DMA controller

A fast MCU was chosen to have a lot of flexibility in the development. ARM cortex M4 and M3 use the same Harvard architecture, which means that we can quite easily calculate if the software can be run on the cheaper M3 family.

The software was developed by using the CooCox CoIDE software development tool with the ARM GNU tool chain. The SW was compiled with the most optimized level (-O3) and FPU was not used. Accuracy tests were run without the FreeRTOS scheduler.

5.2 Test set-up

The accuracy was tested by using a PC simulation and a robot tester (appendix 3). Test parameters were chosen so that the results are comparable. The algorithm was configured to support greater than 4mm size of touches (referring to a perfect circle's diameter) and the runtime calibration was enabled. Touch height, width and pressure were also reported. Touch algorithm resolution was configured to 437 * 683 (X, Y). The actual touch area size is in millimetres 50 * 78 (X, Y), which means that the maximum theoretical accuracy is 0.114 mm.

The robot tester's touch head is an off the shelf component made of silicone. It forms a circle having a diameter of 7mm on the touch panel. The same size of a virtual test finger was used in the touch simulation. There was no accurate positioning method to calibrate the robot test head with the touch panel's coordinate system. For that reason the offset errors were first corrected in the results. Anyhow offsets were very small +-0.3mm.

5.3 Touch accuracy

There are various parameters that describe the touch accuracy: distance error, jitter and linearity. The distance error tells how far the output is from the input. The result is the error vector's length. Min, max and average errors are calculated.

Jitter tells how much touch moves (jitters) when a finger is down and stationary. The maximum jitter is reported for x and y axis separately. 5 measurements were done for each test point.

Linearity tells how much error there is from the linear input. If the input is a straight line on x axis, the maximum error is found on y-axis.

Figure 16 illustrates the simulation results in a coloured map. In this case the simulation tested each possible point on the touch area. Green errors are acceptable, which means errors that are smaller than 1mm. The simulation SW does not produce any noise thus the detected channels are just binary ones or zeros. The MCU software uses the signal levels for a more accurate touch position detection, but it was not fully implemented when tests were performed.

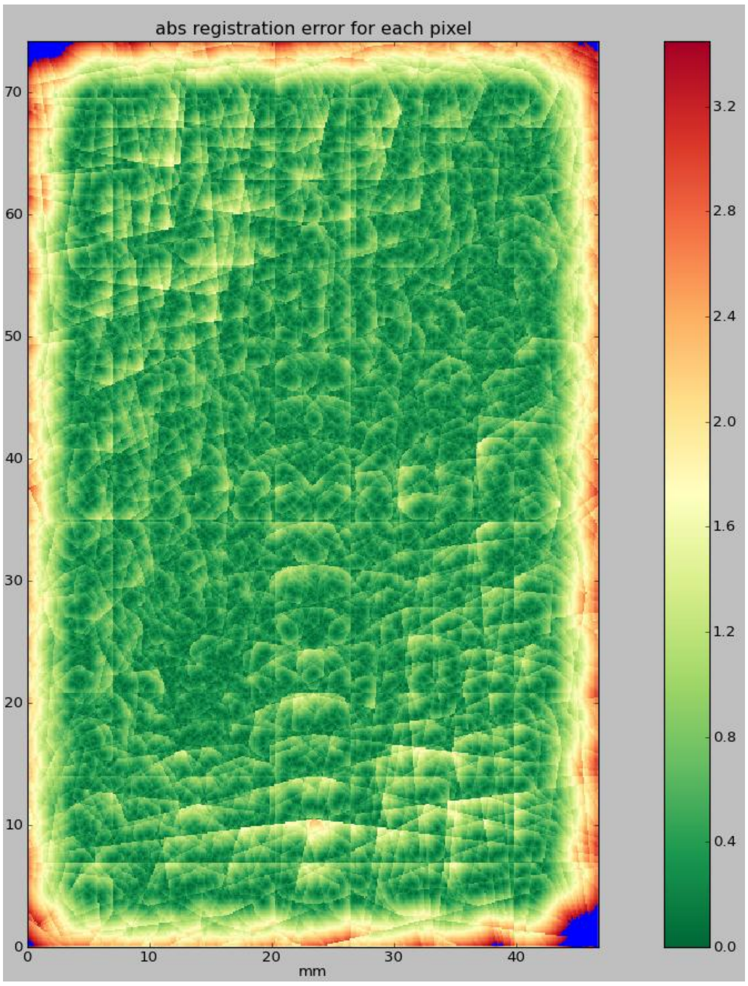


FIGURE 16. Absolute errors in touch area (values in mm)

Various tests were run and the overall result can be seen in the next table (table 5). The errors are reported in millimetres. Appendix 4 shows more detailed results from one test suite. Different tests were run with different sets of input points and with different test area limits. A test area limit means how many per cent of finger's diameter goes outside of the touch area. When a part of the test finger was outside of the touch area, the centre point of the input was calculated from the finger area inside the touch area.

TABLE 5. Comparison of simulation and robot testing

	Simulation worst case (mm)	Robot worst case (mm)
Average distance error	0,64	0,65
Max distance error	2,9	1,9
Linearity error max x	1,4	1,4
Linearity error max y	2,8	1,7
Max jitter		1.5
Offset x		0,2
Offset y		0
Pressure (0-255)		232
Width	9,5	6,4
Height	9,6	5,9

The maximum errors were found near the touch area's edges. Little offset was found on x-axis, the manual robot calibration seemed to be quite exact. The reported pressure was quite high, because silicon is a soft and flat material, and has a high refractive index. There is some error in the width and height values, but this is still in the theoretical range of the maximum error. Why the simulation and the robot testing gave so different results with these attributes needs more investigation.

Figure 17 shows the results of a 10 * 16 tapping test with 160 testing points in total. Some jitter can be seen in some test points. The max jitter in this test was 0.8 mm and the average value was near zero. This is because a very strong jitter filter was used in the algorithm.

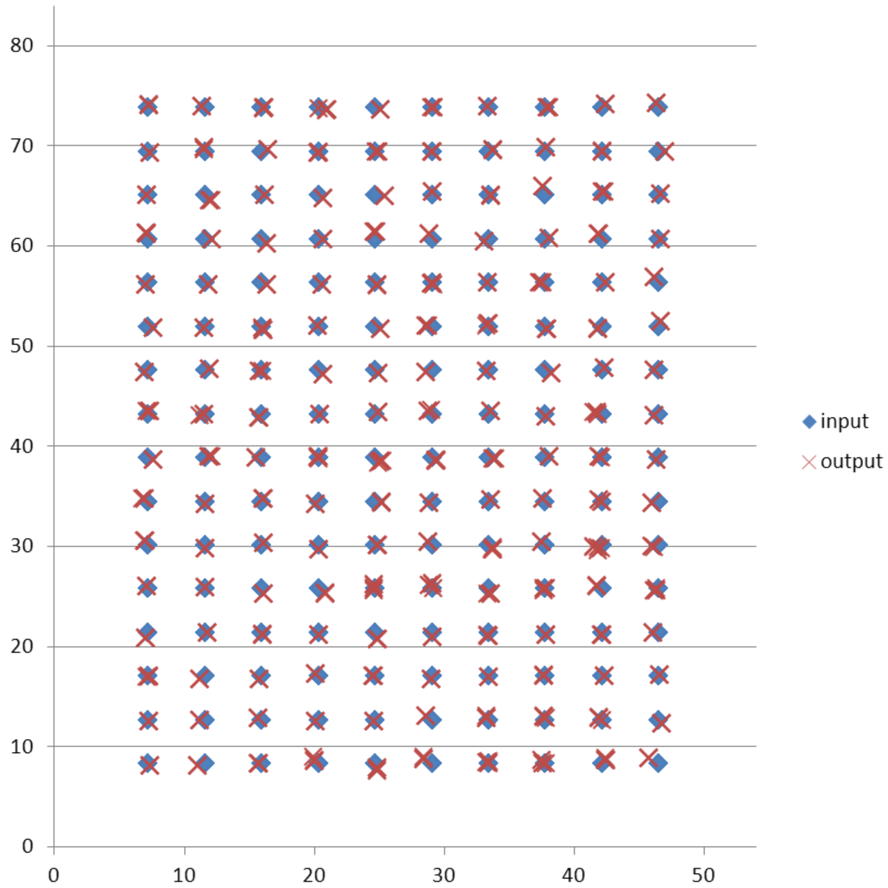


*FIGURE 17. Example of a 10 * 16 test scan (values in mm)*

5.4 MCU Performance

The tested prototype was not configured to give the best possible frame rate performance. The phototransistors and amplifiers were tuned to give the best sensitivity for the touch detection, because the sensitivity is the most critical requirement for the FTIR based touch. The phototransistor response time was quite slow being around 60us.The algorithm performance was measured by using GPIO toggling and oscilloscope. Tests were run with and without the FreeRTOS scheduler and clocking CPU at 168 MHz and 48 MHz.

Table 6 summarizes the performance testing results. There is a small improvement in the frame rate when using the scheduler. This is because the touch panel scanning and algorithm processing can happen virtually concurrently. The context switch from algorithm processing happens every time when the photo receivers scanning timer expires (appendix 1). If the scanning time was much shorter, the difference would be less.

The algorithm processing does not take much time. When running CPU at 48 MHz and using 3 fingers, the algorithm processing time is around 6ms, which would still give a good frame rate for the system if faster photo receivers were used. By increasing the finger count, the algorithm continues to perform well.

TABLE 6. Performance testing results

No scheduler	168	48	MHz
1 touch algo time	0,5	2	ms
3 touch algo time	1,15	6	ms
Frame rate	94	71	Fps
RTOS scheduler			
1 touch algo time	0,55	2,5	ms
3 touch algo time	1,2	5,7	ms
Frame rate	100	83	Fps

Figure 18 shows the elapsed time for each step in the photo receiver reading cycle after LED has been pulsed. It is easy to see that the bottle-neck is in the photo receiver's rise/fall time. Other interesting thing is the digital low pass filter, which is of first degree. It takes 2us to filter 5 samples (5 photo receivers) of 16 bit data. The second and higher order filters would have a significant influence on an overall performance of the system. If heavier digital filtering is needed, a FPU or a specific DSP unit is needed.

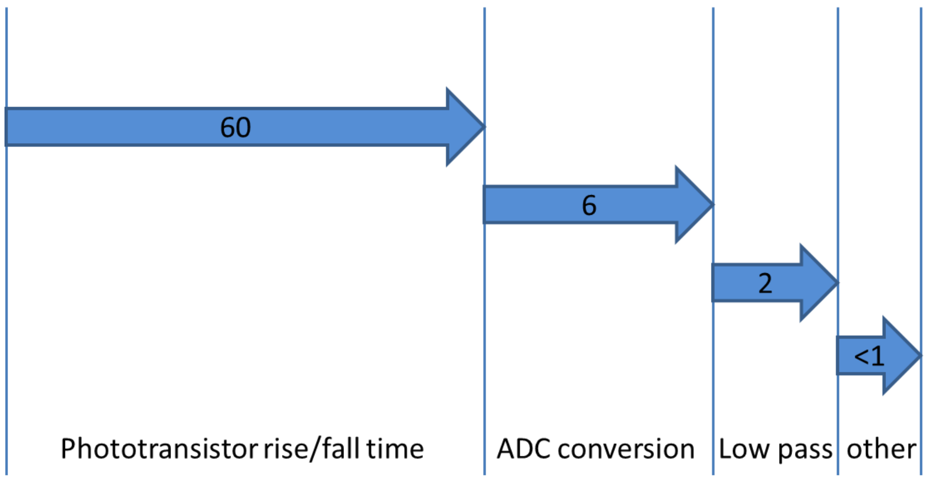


FIGURE 18. Time analysis in photo receiver reading

The dynamic memory consumption was measured by using API functions of FreeRTOS and by observing memory from the debugger. The measured RAM memory consumption was 7Kb of which 500 bytes is occupied by FreeRTOS. The ROM memory consumption was read from a flasher application giving 48 kb of which 6kb is occupied by FreeRTOS. The memory consumption was within the initial estimations.

The power consumption was measured only for LEDs, because the STM32F4 development kit was equipped with many unnecessary peripherals, thus giving greater results. The measured power consumption of LEDS was high, around 20mW. This is because of low response time of phototransistors.

5.5 Overall functionality and conclusions

The performance measurements showed that a mid-range Cortex-M3 MCU (~48MHz) without FPU would meet the performance requirements, with the assumption that no heavy digital filtering is done. The frame rate is at acceptable level around 80 FPS. Most devices in the market have frame rates between 80-120. The performance can be still improved significantly by doing SW optimization.

The measured accuracy is at the same level as in the capacitive touch devices. Referring to a touch screen test in an mPC magazine (*Jääskeläinen & Tervola* 2012, 44-47), on average touch devices had the maximum errors between 1-2.5 mm and the average error below 1 mm. It is important to note that in those tests the testing was done 6 mm from the edges of screen.

The prototype device was configured to support a multi touch up to four fingers. The multi touch functionality was only tested manually. The device performed well with two fingers, but adding more fingers started to cause some errors. Also, leather and rubber gloves were tested successfully with the touch panel, only thick wool gloves were difficult to detect by the system.

The device was tested in the sun light and it saturated quite easily. This is because phototransistors have a narrow dynamic range. It is clear now that phototransistors are not a good choice.

The developed FTIR panel and the controller were integrated into a commercial phone with new mechanics, in order to test the usability aspects. The conclusion was that the sensitivity was not at an acceptable level yet. The prototype was tuned to get the full sensitivity from ADC converters, but the signal noise was too high in order to detect small changes (SNR too high). The phone integration did not show any extra interference in the touch functionality. Appropriate and precise measurements should be done to confirm this.

6 SUMMARY

The project proved that it is possible to implement a multi touch device by using TactoTek's patented FTIR touch. The project started from the algorithm work. First a simulation SW was built to study different algorithm options. A PC testing tool was developed to study signal behavior on a touch panel. After a good algorithm candidate was found, it was integrated into MCU. The integration showed that building a real time system is not straight forward, for example a wrongly implemented scheduling can cause jitter in timer. The developed SW was capable of running with or without a scheduler having no significant jitter in timing.

The initial performance estimates were quite accurate. The algorithm performed well on the MCU development kit at lower frequencies. The memory consumption was also at decent level. Also, it is important to note that the algorithm was not performance and memory optimized. The target was to implement an algorithm that is easy to port and maintain and to support different screen sizes. The developed SW architecture, MCU and algorithm SW gave a good base for further development.

In overall the project went well, a lot of information was gathered and I personally learned a lot about MCU and touch algorithm development, analog and digital hardware and optics. The hardest part was the MCU programming. The latest, just launched, development kit was chosen to have a good platform, but it was difficult to find support and example applications for a new development kit. In the beginning the work was quite algorithm and simulation oriented. One lesson learnt was that the iteration cycle between simulation and using real data must be short. Not all algorithm ideas were easy to implement, neither were they easy to integrate in MCU.

The next step in the development will be to design a new component layout as the tested panel structure showed some weakness in the multi touch performance. Photodiodes will be used instead of phototransistors, because they do not saturate in the sun light and they are much faster. Photodiodes have a wide dynamic range, but they set other challenges for the system. The receiver's signal must be greatly amplified near the photodiode or a very well sheltered signal path to the controller must be implemented, which can be challenging.

Also, a new dedicated MCU board with only the necessary peripherals needs to be developed in order to make a simpler design and to eliminate unnecessary interference sources.

The capacitive technology is dominating the mobile device markets today, thus it sets a benchmark for other technologies. The challenge in the FTIR technology is the activation sensitivity compared to capacitive, which does not need a physical contact for touch activation.

On the other hand the FTIR panel provides a more natural and secure input mechanism for applications where unintentional touches should be avoided. This could be advantageous in consumer applications where usually buttons and switches have been used, for example household and automotive applications. In these markets the game is still widely open and this is where the FTIR application may have a good chance for success. In order to achieve an optimal solution, more studies are needed in the material, HW and signal processing level. To beat the challenge, a top class engineering work is needed.

LIST OF REFERENCES

Benson, H. 1995. University Physics. New York: John Wiley & Sons Inc.

Cohen, R., Small, J. 2013. Lambert’s law. 4physics. Date of retrieval 6.11.2013 http://www.4physics.com/tn3/lambertian.htm

DisplaySearch 2012. Touch Panel Market Analysis. Date of retrieval 5.6.2013 http://www.displaysearch.com/cps/rde/xchg/displaysearch/hs.xsl/120717_tablet_and_smart_phone_demand_to_drive_2012_touch_screen_revenue_to_16_billion.asp

Global Display Engineering 2012. Touch technologies compared. Date of retrieval 8.9.2013 http://globaldisplayengineering.com/Conclusion.php

IDTechEx 2012. Touch market forecast by technology. Date of retrieval 8.9.2013 http://www.printedelectronicsworld.com/articles/touch-screen-market-to-reach-14-billion-in-2012-00004286.asp

Jaine, T. 2011. Technical data sheet of infrared LED. Date of retrieval 6.11.2013 https://www1.elfa.se/data1/wwwroot/assets/datasheets/IR83-01C-TR8_eng_tds.pdf

Jääskeläinen, O., Tervola, J. 2012, Kosketusnäytöt (touch screens), mPC 12/2012, 44-47.

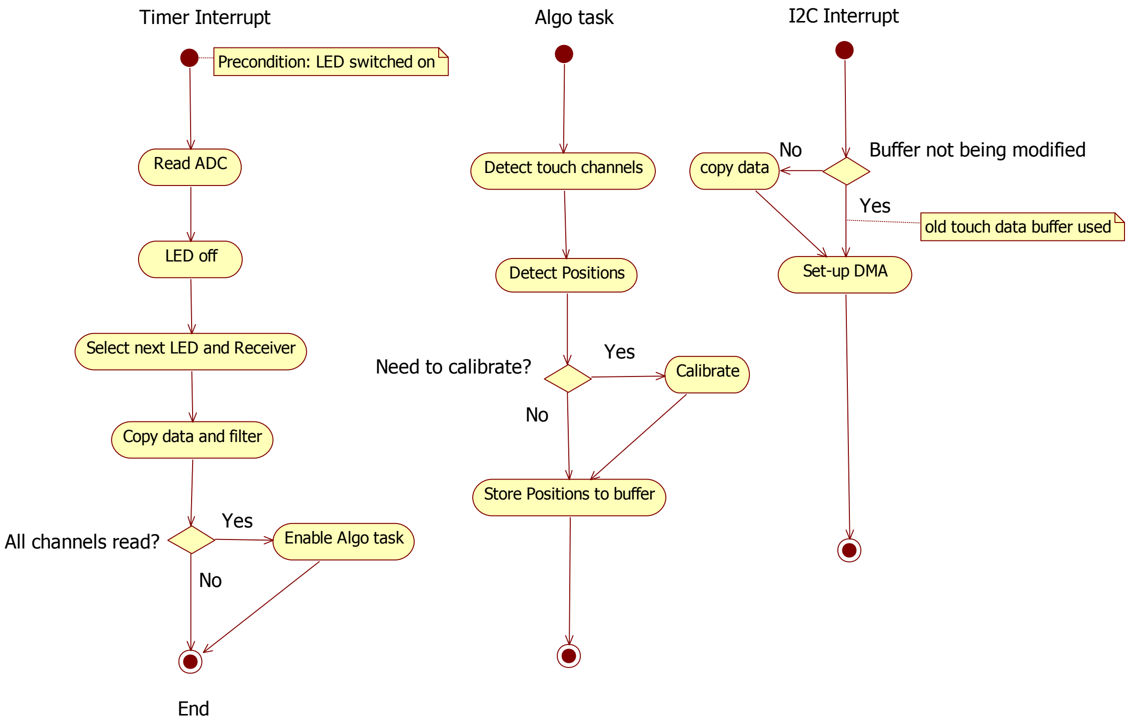
FI 121862B. 2011. Järjestely kosketusnäyttöä varten ja vataava valmistusmenetelmä. Valtion teknillinen tutkimuskeskus, Espoo, Suomi. (Aikio, J. Keränen, A.) 20086009, 24.10.2008. Published 25.04.2010. 24p.

Zhu, S., Yu, A., Hawley, D., Roy R. 1986. Frustrated total internal reflection. Date of retrieval 8.9.2013 http://www.cleyet.org/Misc._Physics/Microwave-Optics/Evanescent-FTIR/FTIR%20review.pdf

APPENDICES

INTERRUPTS AND TASKS

APPENDIX 1

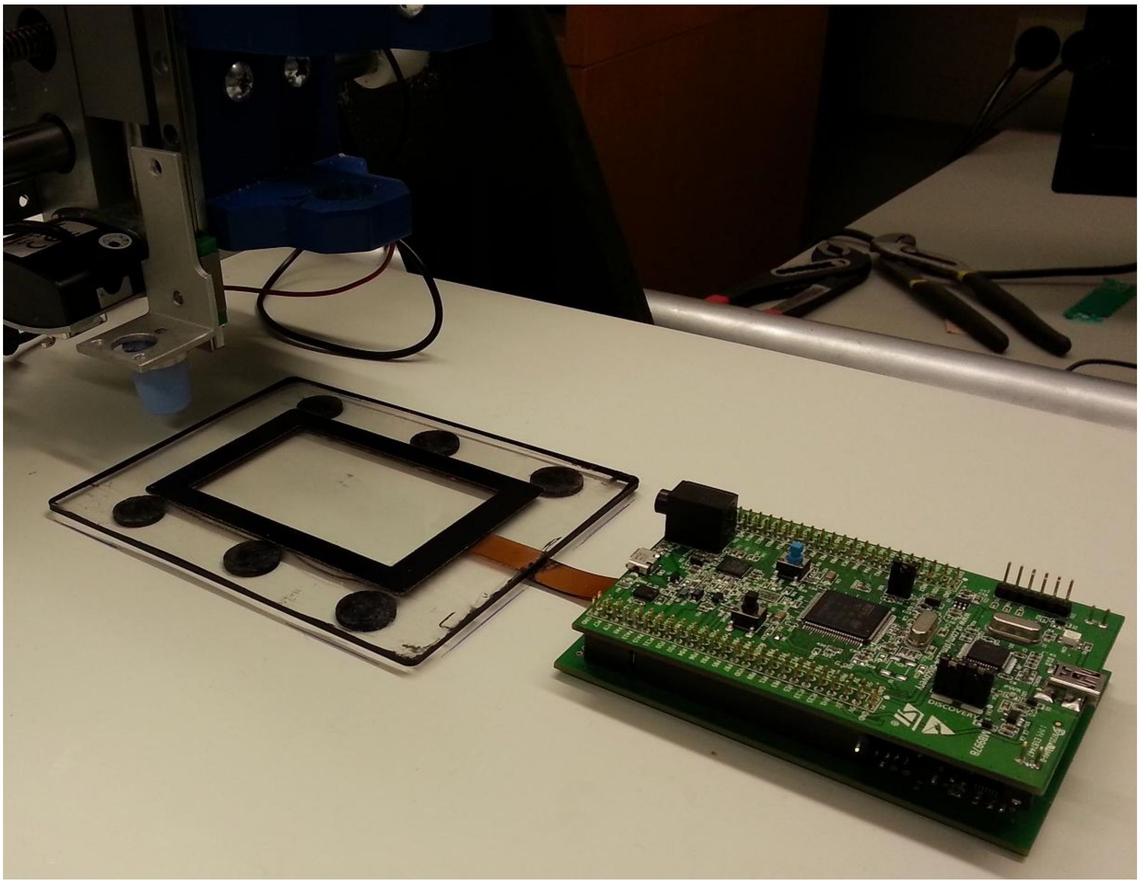


Cortex-M feature set comparison

	Cortex-M0	Cortex-M3	Cortex-M4
Architecture Version	V6M	v7M	v7ME
Instruction set architecture	Thumb, Thumb-2 System Instructions	Thumb + Thumb-2	Thumb + Thumb-2, DSP, SIMD, FP
DMIPS/MHz	0.9	1.25	1.25
Bus interfaces	1	3	3
Integrated NVIC	Yes	Yes	Yes
Number interrupts	1-32 + NMI	1-240 + NMI	1-240 + NMI
Interrupt priorities	4	8-256	8-256
Breakpoints, Watchpoints	4/2/0, 2/1/0	8/4/0, 2/1/0	8/4/0, 2/1/0
Memory Protection Unit (MPU)	No	Yes (Option)	Yes (Option)
Integrated trace option (ETM)	No	Yes (Option)	Yes (Option)
Fault Robust Interface	No	Yes (Option)	No
Single Cycle Multiply	Yes (Option)	Yes	Yes
Hardware Divide	No	Yes	Yes
WIC Support	Yes	Yes	Yes
Bit banding support	No	Yes	Yes
Single cycle DSP/SIMD	No	No	Yes
Floating point hardware	No	No	Yes
Bus protocol	AHB Lite	AHB Lite, APB	AHB Lite, APB
CMSIS Support	Yes	Yes	Yes

STMicroelectronics

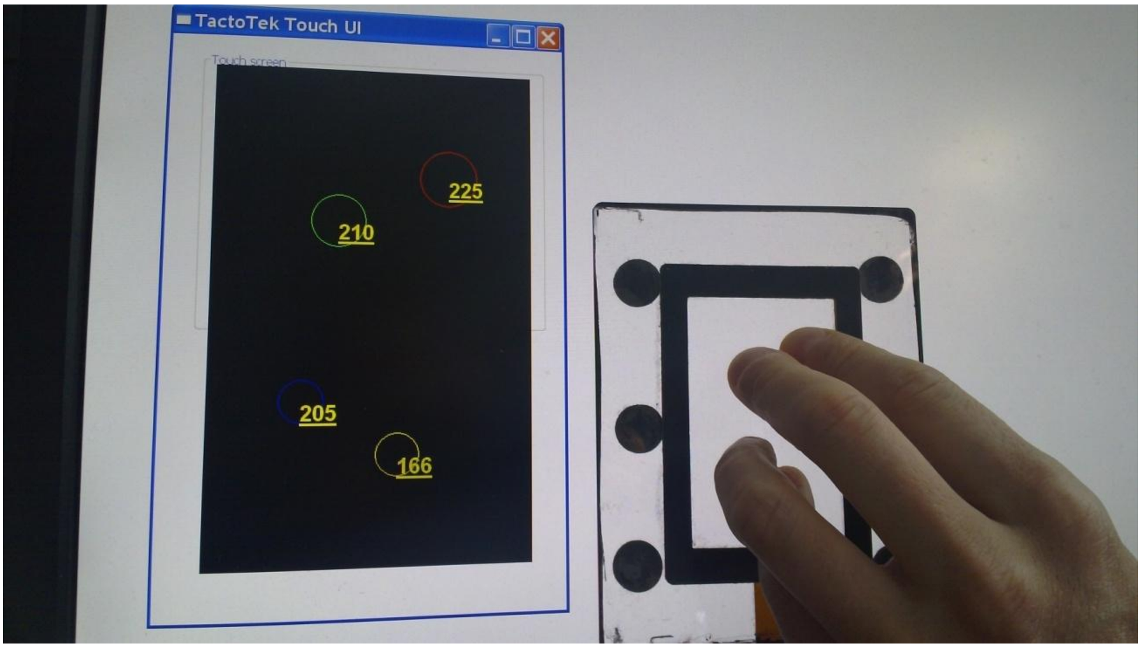
Cortex
Low-Power Leadership from ARM



TESTING RESULTS

APPENDIX 4

	simulation		robot									
test points (x*y)	10*16	border 10*16	10*16	16*10	16*10_45	16*10 border	10*16 border	10*16 border_15	10*16 border_15	10*16 border_20	10*16 border_20	worst case in robot testing
average dist. error	0,5	**0,64**	0,35	0,33	0,33	0,65	0,59	0,4	0,46	0,4	0,46	**0,65**
max dist. error	1,7	**2,9**	0,97	1,1	1	1,6	1,9	1,3	1,5	1,45	1,5	**1,9**
linearity max x	**1,4**	1,3	0,7	1	0,9	1	1	1	1,2	0,8	1,2	**1,4**
linearity max y	1,1	**2,8**	0,8	0,7	0,7	1,3	1,7	1	1,2	0,9	1,1	**1,7**
average offset x			0,2	0,2	0	0,3	0,3	0,2		0,2		**0,2**
average offset x			0	0	0	0	0,1	0,1		0,15		**0**
average pressure			224	235	230	238	235	231		230		**231,8571429**
average width	9,5	8,6	6,4	5,4	6	5,1	5,3	5,7		6		**6,444444444**
average height	9,6	9,1	5,1	4,9	5	4,8	4,9	5		5		**5,933333333**
offset correction for input						x	x		x		x	
% of finger outside						50	50	25	25	16	16	


TactoTek Touch UI
Touch screen
225
210
205
166