



COE ABET COMMITTEE

Activity Report

Term T111

COMPUTER ENGINEERING

Program

at

King Fahd University of Petroleum & Minerals
DHAHRAN, SAUDI ARABIA

Revised May 2012

CONFIDENTIAL

The information supplied in this Self-Study Report is for the confidential use of ABET and its authorized agents, and will not be disclosed without authorization of the institution concerned, except for summary data not identifiable to a specific institution.

Table of Contents

PLANNING AND ASSESSMENT FOR NEXT TWO YEARS	3
CONTINUOUS IMPROVEMENT in T111.....	4
Action to Improve the Program.....	4
Improving Engineering Design.....	6
Improving the Integration of Hardware and Software.....	9
Appendix A. GUIDELINES FOR DEVELOPING PROJECTS.....	13
Appendix B. PROJECTS ON ENGINEERING DESIGN	14
COE 203 : An XO Game Using XILINX FPGA	14
COE 203: Hack-Proof Authentication for RFID Car Immobilizer Key.....	19
COE 203: Design of a Reaction Timer Using FPGA.....	21
COE 305: Real-Time EDGE Detection.....	23
COE 305: Automatic Elevator.....	26
Appendix C. PROJECTS ON HARD/SOFT INTEGRATION.....	28
COE 306: Infrared-Based Home Security System	28
COE 306: VOICE-ENABLED SYSTEMS: INTERFACING AND RELIABILITY	34
COE 306: Voice based Home Navigation System for the Physically Challenged.....	37
COE 306: Unicursal- Gesture Interface for TV Remotes with Touch-pads	41
COE 306: GSM-based Automobile Anti-Theft System.....	46
COE 306: Controlling a 3 Wheel Drive Omni-directional for Line Following	52

PLANNING AND ASSESSMENT FOR NEXT TWO YEARS

The ABET Program Assessment Committee approved the following Action Plan for the following two academic years.

Action Plan

First Year 2010-2011:

1. Addressing the comments raised by ABET in the Final Statement as well as those that were verbally formulated:
 1. Proposing some revisions,
 2. Seeking approval by COE faculty,
 3. Revising the COE the assessment system (rubrics and surveys) based on potential revisions.

2. Develop a policy for the effective implementation of Continuous improvements:
 1. Proposing a policy,
 2. Approval by COE faculty, and
 3. Implementation.

3. Continuous improvement of the COE program, identify programs outcomes that have weak performance indicator using available assessment data, selection of some outcome for improvement, carry out improvement, and documenting.

Second Year 2011-2012:

1. Continuous improvement of the COE program, identify programs outcomes that have weak performance indicator using available assessment data, selection of some outcome for improvement, carry out improvement, and documenting.

2. Develop logistic to address all program outcomes. Provide supporting material.

The above plan can be summarized using the following Table:

COE PLAN	T111	T112	T121	T122
Continuous Improvement	Two Selected POs		Two Selected POs	
Program Assessment		Assessing all Program		Assessing all Program

		outcomes (a)-(k)		outcomes (a)-(k)
--	--	------------------	--	------------------

CONTINUOUS IMPROVEMENT in T111

Continuous improvement of the COE program is a continuous task that is carried out by the curriculum and the ABET Committees. Curriculum revisions or corrective actions proposed by either of the above committees are presented to all COE faculty members in council meetings for discussion, review, and approval.

Action to Improve the Program

The process of direct and indirect assessment started in 2007, it has taken a few years to mature into an acceptable level. Using the collected direct and indirect assessment data, the COE ABET Committee started analyzing the level of achievement of each program outcome based on passed assessment data. The committee has made use of the POs achievement data as the basis for improving some POs in each semester in accordance with the COE Assessment Plan. In T081, the COE ABET committee started implementing the process adopted for continuous improvement in the COE Program.

The POs assessment process uses the following assessment channels: (1) the Exit Exam, (2) the average score in supporting courses, (3) the Exit Survey, (4) the COOP Supervisor and Employer Surveys, and (4) the Rubrics score for a set of representative core junior and senior courses

In Criterion 3, the status of the POs were grouped together in one single assessment table which includes for each PO the results of various assessment channels, the performance criteria, the score representing the level of achievement, and a summary of the committee comments. Following the analysis of the assessment data, the committee adopted a 2-year continuous improvement plan. The committee decided to improve Outcomes (c) and (g-W) in the academic year 2008-2009. These outcomes were selected first because of their importance to the program and due to their low score. These improvements were originally based on developing some guidelines for the students. Over time the issue of Continuous improvement matured into the need to provide teaching material in order to directly improve the student learning. For this reason, the committee carried out Continuous Improvement (CI) for (g-W) and decided to develop appropriate teaching material for COE 390 on three soft outcomes which are the “Engineering Ethics”, “Awareness of Contemporary Issues”, and “Awareness of the Impact of Engineering Solution on Society”. These can be found in our term report:

<http://faculty.kfupm.edu.sa/coe/mayez/ABET/COE-ABET-Committee-2011-2012/COE-ABET-Activity-Report-T102-Oct-2011.doc>

In Term 111, the committee analyzed (1) the previously collected direct and indirect assessment data and (2) current status of the program outcomes. The above issues were largely debated and the committee and the committee decided to carry out CI cycle in T111 for improving the (1) Engineering Design, and (2) the Integration of Hardware and software, an extra outcome added

by the COE faculty to the standard ABET a-k outcomes. On 1st October 2011, the committee assigned following faculty:

- a. **Dr. Yahia Osais**, assistant professor at the COE, was assigned as Faculty in Charge to carry out the CI for the Integration of Hardware and software. On 18 December 2011 Dr. Yahia Osais decided to retire from the committee. No work was delivered for the above task. The Department chairman was informed of the above problem. However, no professorial rank faculty was assigned to substitute the above. On January 10, 2012, Mr. Mohamad Felemban (Lecturer) and Mr. Ali Al-Suwaiyan were added to the committee. In view of the above problem, the committee was in difficulty to deliver the work in a decent way. The reported work is still incomplete and we wish the additional members will help alleviate the above problem. The committee deeply thanks Dr. Wassim Raad for volunteering to help in the above task.
- b. **Dr. Mohamad Sqalli**, assistant professor at the COE, was assigned as Faculty in Charge to carry out the CI for the Integration of Hardware and software. The committee deeply thanks Dr. Wassim Raad for volunteering to help Dr. Sqalli in the above task.

Improving Engineering Design

In T111, following the analysis and examination of the previous Rubric Assessment Data and the available indirect assessment data, the committee decided to carry out Continuous Improvement measures for the Program Outcome on Engineering Design (c) and Dr. Waseem Raad (will be referred to as faculty in charge) has been nominated to conduct the above process. Outcome (c) had always received somehow marginal scores of about 2.6/4, and accordingly was selected for improvement because of its importance to the COE program.

For Program Outcome on Engineering Design (c), the ratings of learning using direct measures with student-by-student. The rubric assessment data is presented as part of the display material. In addition, the display material presents the performance criteria used for the rating of each PO. Outcome “C” has five performance criteria: (1) translate general requirements, (2) identify and formulate any problem, (3) list different design alternatives, (4) choose the appropriate design, (5) fine tune the chosen solution. Average student performance was considered inadequate (score below 2.5 / 4) score for the performance criteria (2), (3), and (4). This indicates that the concept of Engineering Design needs to be better defined, practiced wherever possible, and be integrated as part of all the COE courses with a culminating engineering design experience in the Capstone course.

To close the loop, the faculty proposed the following action plan:

1. In the light of the new COE BSc Program, there is need to spread the education of Engineering Design over all the Program courses. Specifically, at the 203 “Logic Design Lab” and COE 306 “Embedded Systems” are basically developed based on a set of lab experiments. Therefore, the Engineering Design package would improve if Mini-projects are adopted for the above courses.
2. The Committee reviewed the course syllabi of above courses and contacted the course instructors to discuss the idea of Mini-projects which received acceptance from both the instructor and committee.
3. Progressively introducing Engineering Design Education in the COE program Labs. Specifically developing or adapting educational material on Engineering Design for faculty lecturing, students reading, and provide web resource and references. Also, coordinating with the concerned course instructors to emphasize outcome (c) and to provide supporting material. Engineering education must be addressed in all relevant courses and labs of the COE program to better prepare the student to the practice of engineering design.
4. The committee decided to develop a set of Mini-projects and work out the project formulation so that the concept of Engineering Design will be progressively introduced in the above courses. Each Min-Project is to be run for no less than 6 weeks. The students will team out for the project and finally submit a report. Having a set of Mini-projects provides the instructor different alternatives to choose from as well as to avoid centering the course for one specific project.
5. Also instructors, may deliver presentations to students aimed at improving outcome (c) “Engineering Design...” to improve their awareness about this issue.

6. Develop templates and guidelines for the students on Engineering Design.
7. Explore how Engineering Design Education can be enhanced in the COE Program.

The following action steps were carried out:

1. Developing the following set of Mini-Projects (See the Appendix A):

- a. XO-Game (COE 203),
- b. Reaction Timer (COE 203),
- c. RFID Authentication (COE 203),
- d. Traffic Light (COE 203),
- e. Elevator (COE 306),
- f. Edge Detection (COE 306).

2. To improve awareness of the students, a detailed description on **Engineering Design Processes** is provided to the students to reflect on its steps and their meaning on their project:

<http://faculty.kfupm.edu.sa/COE/mayez/ABET/COE-ABET-Committee-2011-2012/Engienering-Design-Process.pdf>

3. **Extending the Engineering Education to COE Labs:** To develop a culture of Engineering Education at the COE, the engineering design concepts must be progressively implemented at different program levels. The committee reiterates its earlier recommendation to introduce some engineering design concepts at all levels in the Programs. Following Table provides a Plan for the Progressive Introduction of Engineering Design (Design a System, Component, or Process) in the COE program. It is recommended to adopt the following behavioral approach in all the COE Labs.

Plan for Progressive Introduction of Engineering Design

Elements of Engineering Design	200-level Courses	300-level Courses	400-level Courses
1. Identifying a need (There is a need for solving this problem!, what is this need?)			Identify the need for solving the problem and domain of application
2. Defining the problem (The design task will be completed if I solve a specific problem, What is this problem)	Word definition of the problem.	Word definition of the problem. Enforcing use of Notebooks in all activities.	Word definition of the problem. Enforcing the use of Notebooks in all activities.
3. Conducting research (How and where to search the library, ebooks, datasheets, manufacturer web etc to find relevant material to solve the problem)			Searching similar problems (wording) and proposed solutions

4. Narrowing the research (knowing the problem to be solved I need to filter Software/Hardware components that may help in building a solution)			Knowing relevant examples be able to identify a few similar systems
5. Analyzing set criteria (knowing the problem, its specifications, and its relevant components found, I need to find out whether some components meet the problem specifications or not)		Analysis of problem specifications and identify components meeting some of the above	Analysis of problem specifications and identify components meeting some of the above
6. Finding alternative solutions (determine a few possible solutions using found components for the above problem, I need to identify each of these possible solutions),	Using different approaches or components. Use of Notebooks.	Combining components to find different solutions.	Combining components to find different solutions
7. Analyzing possible solutions (knowing the problem (specifications) and possible solutions I need to find out whether some solutions meets the problem specifications or not)		Pruning possible solutions which do not meet the specifications	Pruning possible solutions which do not meet the specifications
8. Making a decision (given two or more possible solutions I need to select one feasible and economical solution)	Selecting an economical solution	Selecting the most economical feasible solution	Selecting the most economical feasible solution (ethics, environmental)
9. Presenting the product (I need to describe my solution in writing using diagrams, graphics, drawings, etc.)			Technical description of the design
10. Communicating (prepare a written report in which each task is broken down into: Identifying a need, Defining the problem, Conducting research, Narrowing down the research, Analyzing set criteria, Finding alternative solutions, Analyzing possible solutions, and Making a decision.)	A simple report describing the above steps.	A report defining the problem, specification, analysis of components vs specification, solutions, solution analysis, and decision	A report defining the problem, specification, analysis of components vs specification, solutions, solution analysis, and decision

Improving the Integration of Hardware and Software

Continuous improvement (CI) is the process of devising and implementing effective corrective actions (CAs) on COE courses and labs to improve on the fulfillment of program outcomes in response to shortcomings detected through ABET assessments. In 2010-2011, the COE ABET Committee developed a framework for integrating Continuous Improvement into the teaching process.

In T111, following the analysis and examination of the previous Rubric Assessment Data and the available indirect assessment data, the committee decided to carry out Continuous Improvement measures for the Program Outcome on Integration of Hardware/Software, i.e., PO (l) (Note that in the SSR, this PO is referred to as PO (n), however, since two POs have been removed, we had to rename this one). This section outlines the proposed continuous improvement for the Hardware and Software Integration in the COE Department

Based on the rubrics assessment through COE 400, COE 485, and COE 351, PO (l) received a score of 2.77 in T062, 071, 072 and a score of 2.68 in T081, which are marginal scores, and accordingly was selected for improvement because of its importance to the COE program. As stated in the program outcomes performance indicators document, Table 1 shows the rubrics that have been used for the Program Outcome on Integration of Hardware/Software, i.e., PO (l).

Although, it is stated in the SSR that three courses are used for assessment of PO (l), i.e., COE 400, COE 485, and COE 351, it has been agreed to change this in the new balancing of the COE Rubric Assessment Process approved by the COE council meeting. In the new balancing, only COE 400 will be used for assessing PO (l) on Hardware/Software integration through student course projects. However, PO (l) will not be assessed in COE 351 and COE 485 because it has been found that this outcome is presently not addressed in these courses. In the newly approved COE B.Sc. curriculum, a newly proposed course on Introduction to Embedded Systems, i.e., COE 305, will cover the aspect of Hardware/Software Integration; and therefore, we propose to incorporate our proposed change related to this continuous improvement in this course. One major proposed change is the adoption of a mini-project in COE 305.

Table 1. Hardware/Software Integration Rubrics

Outcome	1 Exemplary	2 Proficient	3 Apprentice	4 Novice
Selection of hardware equipment and software	Selection of hardware equipment and software follows a thorough approach where many criterions are used: performance,	Selection of hardware equipment and software follows a thorough approach where few criterions are used: performance,	Selection of hardware equipment and software is based on the selection of a single manufacturer	Selection of hardware equipment and software is based on the suggestions

	compatibility, standard compliance, protocol support, interoperability, manufacturer strength.	compatibility, standard compliance. One or more relevant important criterions are ignored.	already integrated solution among several candidates.	of the marketing team of one single vendor
Integration Methodology	The integration methodology is well described and followed. Interfaces are well defined and their compatibility discussed. Use of an integration plan featuring integration phases and a test plan for each phase.	The integration methodology is well described and followed. Interfaces are mentioned but their compatibility is not considered. No use of an integration plan. Some mention of a test plan.	The integration methodology is not described properly and not always followed. Interfaces are not mentioned. No use of an integration plan. Tests are carried out without a plan.	An ad-hoc integration (No) methodology is followed but not described. No use of an integration plan. Tests are carried out without a plan.

Assessment Plan

The following action plan has been proposed for CI of PO (1):

- Review the current status of PO (1) including delivery material (Lab, course project, etc.), and point to strengths, and weaknesses.
- Discuss with relevant COE faculty members how to improve the students' knowledge and skills with respect to PO (1).
- Benchmarking and review of similar experiences in other universities.
- Set a target for what to improve in terms of specific curricular actions. We can make use of the approved guidelines for CI, and whether there is a need to develop/adapt teaching material, presentations, guidelines, template, as well as behavioral aspect like students work organization in the lab and method of student working.
- Discuss the proposed curricular actions with relevant COE faculty members.
- Update the proposed curricular actions based on feedback obtained.
- Present the proposed curricular actions to the ABET committee for further discussion and refinements.
- Present the proposed curricular actions to the COE Council for discussion and approval.
- Report the approved curricular actions onto the COE Web-Syllabi.

Scope

The CI process aims at improving the fulfillment of Program Outcome (1) on Hardware/Software Integration of the COE program, which needs improvement and its scope covers the COE-305 course and lab that contribute to this outcome.

Procedure

For the purpose of improving the PO (1), the committee decided to develop a set of Mini-projects so that the concept of integration of hardware and software, i.e., PO (1) is introduced in the COE 305 course. Each Min-Project is to be run for no less than 6 weeks. The students will team out for the project and finally submit a report. Having a set of Mini-projects provides the instructor with different alternatives to choose from. In addition, the committee recommends the implementation of the agreed corrective action plan by the concerned faculty members teaching the COE 305 course.

Proposed Mini Projects for COE-305

Table 2 presents a list of the proposed Papers that can be used to build Mini-Projects for COE-305 (developed with assistance of Dr. Wassim Raad). More details are being worked out for this list of projects.

Table 2. Proposed Mini Projects for COE-305

Paper Title	Interdisciplinary Nature	Level of Difficulty	Novelty	Student Interest	Hardware Availability	Information Available about Implementation
Automated Voice based Home Navigation System for the Elderly and the Physically Challenged	High	Medium	High	High	Not readily	Medium
Design and Implementation of Pyroelectric Infrared Sensor Based Security System Using Microcontroller	Low	Low-Medium	Low	Medium	Yes	Medium
Development of an Electronic Stick with Audio Sensory Perception for the Visually Impaired	Medium	Medium	Medium	High	Yes	High
A Novel Light-Sensor-Based Information Transmission System for Indoor Positioning and Navigation	Low	High	High	High	Not readily	Low
Electronic Speaking Glove for Speechless Patients - A Tongue to a Dumb	High	High	High	High	Not readily	Low
Automobile Anti-theft System Design based on GSM	Low	Medium	Medium	High	Not readily	Medium
Design of Intelligent Fire Alarm System Based on GSM Network	Low	Low-Medium	Low	Low	Yes, mostly	Low

Outcomes from the CI for the Integration of Hardware and Software

No Mini-Project outcomes were reported by the faculty in charge. In view of this difficulty, the ABET committee decided to Expand the CI cycle to T112 through assigning the newly appointed members (lecturers and RAs) to develop Mini-Projects for addressing the above program outcome.

Appendix A. GUIDELINES FOR DEVELOPING PROJECTS

The ABET Committee is currently carrying Continuous Improvement for its Program outcomes: (1) Engineering Design, and (2) Integration of Hardware and Software. One of the tasks is to develop Mini-Projects for COE 305 as a set of curricular actions to address the above issues.

Therefore, the objective of the ABET committee is to develop a set of Mini-Projects for the course COE 305: Embedded Systems (3-3-4) (see attached syllabi). This course uses the Arm-Processor as the basic embedded microprocessor.

Each committee member is provided with two papers which are basically published student projects. Each paper will serve as the basis for developing a 6-week Min-Project that can be used the students.

What Min-Project are to be developed for COE 305 Embedded Systems

Normally the instructor selects one Min-Project for all the class starting from the 8th week of the semester. The students will be provided with a write-up describing the details of the Mini-Project and needed hardware interfacing and associated software. Subsequently, the instructor describes the Mini-project and supervises the student work in each 3-hour weekly lab in each of the remaining 6-weeks. The student will be supported with the needed hardware interfacing as the instructor is likely to plan ahead of time the specific Mini-Project for the semester.

Your task is to read the attached paper and write up a Mini-Project based on the described in the paper. To be **used by the student**, we need the following format and guidelines:

1. Have a title for the Mini-project
2. A summary of the project describing the project idea and its application or use
3. A detailed description in which we develop the:
 - a. The detailed project idea,
 - b. Interfacing: the Mini-project implementation aspects or interfacing to the microprocessor. A schematic with its description and interfacing to microprocessor (just indicate ARM processor) with specification of the components or hardware to be used. Some textual description on how the schematic works, or how it should be programmed to operate in a proper way.
4. A detailed description of the application program to be developed by the student in order implement the project idea described above. How the system can be operated and tested

by the student (based on project idea). Some interesting project aspects (if any) that needs to be experimented. Finally, how the project can be improved (if any).

The deadline for receiving the above 2 Min-project descriptions is January 18, 2012.

Appendix B. PROJECTS ON ENGINEERING DESIGN

COE 203 : An XO Game Using XILINX FPGA

Thanks to Dr. Wassim Raad for providing this material

The project is based on designing & building an XO game for 2 players using a XILINX FPGA and external decoders & colourful LEDs.

An array of 3X3 LEDs will be built on a separate board to mimic the XO game, such that it will take 1 player against the machine player. Each of the players will have one LED with different colour merged into same location on the XO matrix. For each location there will be a unique 4-bit binary number. The computer has to start the game using randomness(Pseudo random numbers) i.e choosing the next hit in a smart way. The winning player will be shown on the LCD and there will also be a timeout if somebody is too slow(out of the game). The decoder will be used externally to cut down the number of outputs going out of the FPGA board. Explore all design techniques to generate the random delay and the computer strategy to play the game.

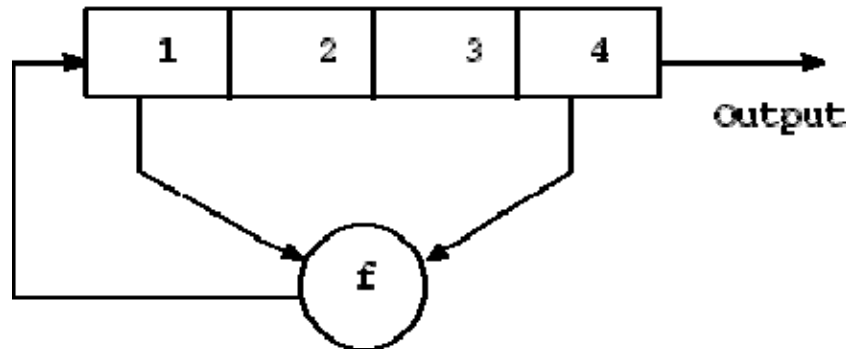
Hint: Use the LFSR (linear feedback shift registers) to generate the random numbers. Tutorial documentation is shown below.

Note: you can find on webct a verilog code for displaying characters on LCD.

Appendix: Linear Feedback Shift Registers

For those unfamiliar with Linear Feedback Shift Registers, see the Primer following Figure 1. For quick information on any of the LFSR parts (shift register, feedback function, output stream, tap sequences) click on the corresponding part in Figure 1.

Figure 1) 4-Bit LFSR, Tap Sequence; [4,1]



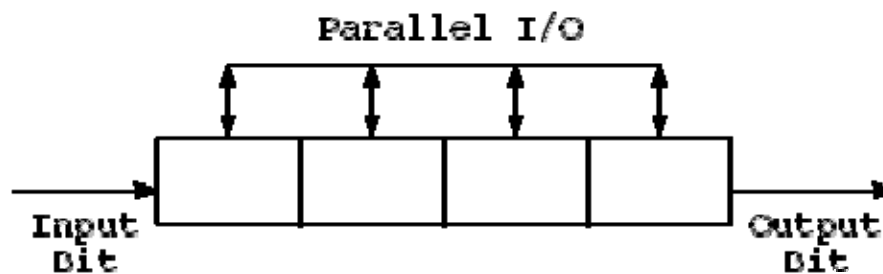
SHIFT REGISTER

One of the two main parts of an LFSR is the shift register (the other being the feedback function). A shift register is a device whose identifying function is to shift its contents into adjacent positions within the register or, in the case of the position on the end, out of the register. The position on the other end is left empty unless some new content is shifted into the register.

The contents of a shift register are usually thought of as being binary, that is, ones and zeroes. If a shift register contains the bit pattern 1101, a shift (to the right in this case) would result in the contents being 0110; another shift yields 0011. After two more shifts, things tend to get boring since the shift register will never contain anything other than zeroes.

Two uses for a shift register are 1) convert between parallel and serial data and 2) delay a serial bit stream. The conversion function can go either way -- fill the shift register positions all at once (parallel) and then shift them out (serial) or shift the contents into the register bit by bit (serial) and then read the contents after the register is full (parallel). The delay function simply shifts the bits from one end of the shift register to the other, providing a delay equal to the length of the shift register.

Figure 2) Shift Register



Some nomenclature:

Clocking) One of the inputs to a shift register is the clock; a shift occurs in the register when this clock input changes state from one to zero (or from zero to one, depending on the implementation). From this, the term "clocking" has arisen to mean activating a shift of the register. Sometimes the register is said to be "strobed" to cause the shift.

Shift direction) A shift register can shift its contents in either direction depending on how the device is designed. (Some registers have extra inputs that dictate the direction of the shift.) For the purposes of this discussion, the shift direction will always be from left to right.

Output) During a shift, the bit on the far right end of the shift register is moved out of the register. This end bit position is often referred to as the output bit. To confuse matters a bit, the bits that are shifted out of the register are also often referred to as output bits. To really muddy the waters, every bit in the shift register is considered to be output during a serial to parallel conversion. Happily, the context in which the term "output" is used generally clears things up.

Input) After a shift, the bit on the left end of the shift register is left empty unless a new bit (one not contained in the original contents) is put into it. This bit is sometimes referred to as the input bit. As with the output bit, there are several different references to input that are clarified by context.

FEEDBACK FUNCTION

In an LFSR, the bits contained in selected positions in the shift register are combined in some sort of function and the result is fed back into the register's input bit. By definition, the selected bit values are collected before the register is clocked and the result of the feedback function is inserted into the shift register during the shift, filling the position that is emptied as a result of the shift.

The feedback function in an LFSR has several names: XOR, odd parity, sum modulo 2. Whatever the name, the function is simple: 1) Add the selected bit values, 2) If the sum is odd, the output of the function is one; otherwise the output is zero. Table 1 shows the output for a 3 input XOR function.

Table 1) XOR Function

Input A	Input B	Input C	XOR Output
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

The bit positions selected for use in the feedback function are called "taps". The list of the taps is known as the "tap sequence". By convention, the output bit of an LFSR that is n bits long is the nth bit; the input bit of an LFSR is bit 1.

TAP SEQUENCES

An LFSR is one of a class of devices known as state machines. The contents of the register, the bits tapped for the feedback function, and the output of the feedback function together describe the state of the LFSR. With each shift, the LFSR moves to a new state. (There is one exception to this -- when the contents of the register are all zeroes, the LFSR will never change state.) For any given state, there can be only one succeeding state. The reverse is also true: any given state can have only one preceding state. For the rest of this discussion, only the contents of the register will be used to describe the state of the LFSR.

A state space of an LFSR is the list of all the states the LFSR can be in for a particular tap sequence and a particular starting value. Any tap sequence will yield at least two state spaces for an LFSR. (One of these spaces will be the one that contains only one state -- the all zero one.) Tap sequences that yield only two state spaces are referred to as maximal length tap sequences.

The state of an LFSR that is n bits long can be any one of 2^n different values. The largest state space possible for such an LFSR will be $2^n - 1$ (all possible values minus the zero state). Because each state can have only one succeeding state, an LFSR with a maximal length tap sequence will pass through every non-zero state once and only once before repeating a state.

One corollary to this behavior is the output bit stream. The period of an LFSR is defined as the length of the stream before it repeats. The period, like the state space, is tied to the tap sequence and the starting value. As a matter of fact, the period is equal to the size of the state space. The longest period possible corresponds to the largest possible state space, which is produced by a maximal length tap sequence. (Hence "maximal length")

Table 2 is a listing of the internal states and the output bit stream of a 4-bit LFSR with tap sequence [4, 1]. (This is the LFSR shown in Figure 1.)

Table 2) 4-Bit LFSR [4, 1] States and Output

Register States				Output Stream
Bit 1 (Tap)	Bit 2	Bit 3	Bit 4 (Tap)	
1	1	0	1	
0	1	1	0	1
0	0	1	1	0
1	0	0	1	1
0	1	0	0	1
0	0	1	0	0
0	0	0	1	0
1	0	0	0	1

1	1	0	0	0
1	1	1	0	0
1	1	1	1	0
0	1	1	1	1
1	0	1	1	1
0	1	0	1	1
1	0	1	0	1
1	1	0	1	0

MAXIMAL LENGTH TAP SEQUENCES

LFSR's can have multiple maximal length tap sequences. A maximal length tap sequence also describes the exponents in what is known as a primitive polynomial mod 2. For example, a tap sequence of 4, 1 describes the primitive polynomial $x^4 + x^1 + 1$. Finding a primitive polynomial mod 2 of degree n (the largest exponent in the polynomial) will yield a maximal length tap sequence for an LFSR that is n bits long.

There is no quick way to determine if a tap sequence is maximal length. However, there are some ways to tell if one is not maximal length:

- 1) Maximal length tap sequences always have an even number of taps.
- 2) The tap values in a maximal length tap sequence are all relatively prime. A tap sequence like 12, 9, 6, 3 will not be maximal length because the tap values are all divisible by 3.

Discovering one maximal length tap sequence leads automatically to another. If a maximal length tap sequence is described by [n, A, B, C], another maximal length tap sequence will be described by [n, n-C, n-B, n-A]. Thus, if [32, 3, 2, 1] is a maximal length tap sequence, [32, 31, 30, 29] will also be a maximal length tap sequence. An interesting behavior of two such tap sequences is that the output bit streams are mirror images in time.

CHARACTERISTICS OF OUTPUT STREAM

By definition, the period of an LFSR is the length of the output stream before it repeats. Besides being non-repetitive, a period of a maximal length stream has other features that are characteristic of random streams.

- 1) Sums of ones and zeroes. In one period of a maximal length stream, the sum of all ones will be one greater than the sum of all zeroes. In a random stream, the difference between the two sums will tend to grow progressively smaller in proportion to the length of the stream as the stream gets longer. In an infinite random stream, the sums will be equal.

2) Runs of ones and zeroes. A run is a pattern of equal values in the bit stream. A bit stream like 10110100 has six runs of the following lengths in order: 1, 1, 2, 1, 1, 2. One period of an n-bit LFSR with a maximal length tap sequence will have $2^{(n-1)}$ runs (e.g., a 5 bit device yields 16 runs in one period). $1/2$ the runs will be one bit long, $1/4$ the runs will be 2 bits long, $1/8$ the runs will be 3 bits long, etc., up to a single run of zeroes that is n-1 bits long and a single run of ones that is n bits long. A random stream of sufficient length shows similar behavior statistically.

3) Shifted stream. Take the stream of bits in one period of an LFSR with a maximal length tap sequence and circularly shift it any number of bits less than the total length. Do a bitwise XOR with the original stream. The resulting pattern will exhibit the behaviors discussed in items 1 and 2. A random stream also shows this behavior.

One characteristic of the LFSR output not shared with a random stream is that the LFSR stream is deterministic. Given knowledge of the present state of the LFSR, the next state can always be predicted.

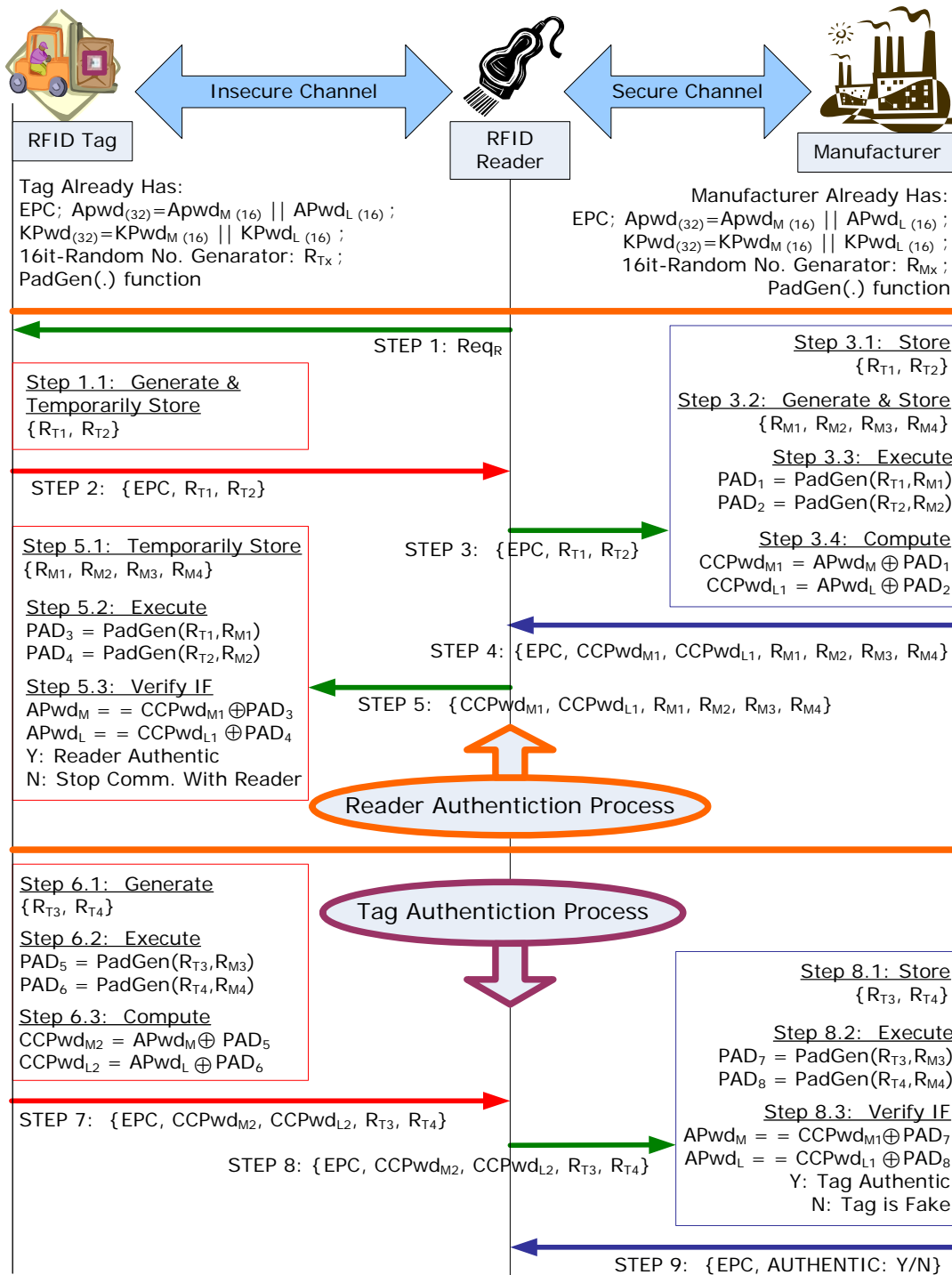
COE 203: Hack-Proof Authentication for RFID Car Immobilizer Key

Thanks to Dr. Wassim Raad for providing this material

The project aims at following the authentication protocols distributed in class between the RFID key & reader. The first stage is to implement the simple protocol between tag & reader, then go to more secure one adding the manufacturer level.

Generate pseudo random numbers using the LFSR (linear feedback shift register) mechanism. Recommended to use statecad for implementing finite state machine for control unit. Each tag & reader share a 32 bit key or password.

At the end of project each group should successfully demonstrate their project documented with a short report. The student should explore all engineering tradeoffs between performance & security, and explore all various alternative solutions for generating random numbers.



COE 203: Design of a Reaction Timer Using FPGA

Thanks to Dr. Wassim Raad for providing this material

Eye-hand coordination is the ability of the eyes and hands to work together to perform a task. A reaction timer circuit measures how fast a human hand can respond after a person sees visual stimulus.

1-The circuit has three input pushbuttons or micro switches, correspond to clear, start & stop signals. It uses a single discrete LED & displays relevant information on a seven segment LED display.

2- A user pushes clear button to force the circuit to return to the initial state and display welcome message the goes off.

3- When ready, a user pushes start button or micro switch.

4- After a random interval from 2-15 seconds, the LED goes on & timer starts counting upward and increases every millisecond and 0.000 displayed on seven segment display.

5- After LED on, user should try to push stop button or micro switch as soon as possible then timer pauses and seven segment display shows reaction time. For most people it should be between 0.15 to 0.3 sec.

6. if stop is not pushed timer stops after 1 second and displays 1.000.

7. if stop button is pushed before LED goes on, it displays 9.999 and stops.

Implement the following:

Derive control & data units and implement design on FPGA.

Hint: use LFSR (Linear feedback shift registers) to generate random number.

COE 203: Traffic Directing Light System

The traffic light (TL) is composed of four faces A, A', B and B' which are intended for use at a four way intersection, where A and A' are two bidirectional roads where A is opposing A'. The same is true for B and B'. Each face is composed of two sets of lights: an 'X' for stop and two connected arrows pointing in the forward and right directions (no left turns for this intersection). The lighting is a set of LEDs, red for the X and green for the arrows.

- The main option for this traffic light is denoted by “Go forward or turn right” which is described below:
 - Enable at time only one forward direction such as A at a time which implicitly means that both the forward AF and turn-right AR directions are enabled. Note that here we may also enable the turn-right of the opposite direction BR because there is no turn left for the above TL.
 - Therefore, the time is split into four parts and each part is allocated to enabling one of four possible states: (AF, AR, and BR), (BF, BR, and AR), (A'F, A'R, and B'R), (B'F, B'R, and A'R) and repeating the above pattern in a round robin fashion.
- The above option can be improved by finding some opportunities to maximize the possibility of non-conflicting turn-rights. Determine this opportunity and rewrite the possible states.
- Now suppose that turn left is allowed for traffic when the forward direction is enabled. In this case, we need to set up the new rule for this traffic light.

Implement the above traffic light of the FPGA board and experience its operation by assuming that a light can be enabled for at most 1 minute. For this you need to use an embedded timer. A normal transition from the green light to the red light must be preceded by a yellow light (warning) for 3 seconds. Include your own analysis for the detection of erroneous functioning and propose a strategy to be used in this case.

COE 306: Real-Time EDGE Detection

The project aims at implementing a real-time Sobel Edge Detection (SED) to motivate the students about computer vision and digital image processing. Edge detection is extensively used in image segmentation to divide an image into areas corresponding to different objects. Edges occur in parts of the image with strong intensity contrast, which often represent object boundaries. Edges can be detected by applying a high pass frequency filter in the Fourier domain or by convolving the image with an appropriate kernel in the spatial domain. Edge detection is commonly performed in the spatial domain, because it is computationally less expensive and often yields better results. Since edges correspond to strong illumination gradients, the derivatives of the image are used for calculating the edges.

The basic edge-detection operator is a matrix area gradient operation that determines the level of variance between different pixels. The edge-detection operator is calculated by forming a matrix centered on a pixel chosen as the center of the matrix area. If the gradient value of this matrix area is above a given threshold, the middle pixel is classified as an edge. Figure 1 shows the setup of the project.

High Level Design

The camera module captures the image and the raw image data is converted into an RGB color space. The image is then converted to grayscale to obtain the image intensity for edge detection. The grayscale image is mirrored to be displayed properly, and stored in the SDRAM FIFO. Three rows of 640-wide pixels are continuously stored as blocks, and edge detection is initiated once the data is ready.

In digital image processing, each image is quantized into pixels. The first step in edge detection is to convert the raw data to a grayscale image, where each pixel indicates the level of brightness of the image: from 0 representing black to 1023 representing white, with a 10-bit wide pixel. The image is then threshold-ed to create a clear gradient. Edge information for a particular pixel is obtained by exploring the brightness of pixels in its neighborhood. If all of the pixels in the neighborhood have the same brightness, it indicates that there is no edge in the area. However, if some of the neighbors are much brighter than the others, it indicates that there is an edge present.

Measuring the relative brightness of pixels in a neighborhood is mathematically analogous to calculating the derivative of brightness. The SED algorithm uses a 3×3 convolution table to store a pixel and its neighbors to calculate the derivatives. The table is moved across the image, pixel by pixel. For a 640×480 image, the convolution table will move through 302964 (638×478) different locations because we cannot calculate the derivative for pixels on the perimeter of the image.

The SED algorithm identifies both the presence of an edge and the direction of the edge. There are eight possible directions: north, northeast, east, southeast, south, southwest, west, and northwest.

For a convolution table, calculating the presence and direction of an edge and is done in three major steps:

1. Calculate the derivative along each of the four orientations. The equations for the derivatives are written in terms of elements of a 3x3 table.

$$\begin{aligned} \text{DerivNE_SW} &= (\text{table}[0,1] + 2 * \text{table}[0,2] + \text{table}[1,2]) \\ &(\text{table}[1,0] + 2 * \text{table}[2,0] + \text{table}[2,1]) \\ \text{DerivN_S} &= (\text{table}[0,0] + 2 * \text{table}[0,1] + \text{table}[0,2]) - \\ &(\text{table}[2,0] + 2 * \text{table}[2,1] + \text{table}[2,2]) \\ \text{DerivE_W} &= (\text{table}[0,2] + 2 * \text{table}[1,2] + \text{table}[2,2]) - \\ &(\text{table}[0,0] + 2 * \text{table}[1,0] + \text{table}[2,0]) \\ \text{DerivNW_SE} &= (\text{table}[1,0] + 2 * \text{table}[0,0] + \text{table}[0,1]) - \\ &(\text{table}[2,1] + 2 * \text{table}[2,2] + \text{table}[1,2]) \end{aligned}$$

2. Find the value and direction of the maximum derivative, and the absolute value of the derivative that is perpendicular to the maximum derivative.

EdgeMax = Maximum of absolute values of four derivatives

DirMax = Direction of EdgeMax

EdgePerp = Absolute value of derivative of direction perpendicular to DirMax

3. Check if the maximum derivative is above the threshold. When comparing the maximum derivative to the threshold, the Sobel algorithm takes into account both the maximum derivative and the derivative in the perpendicular direction.

```
if EdgeMax + EdgePerp/8 >= threshold then
    Edge = true
    Dir = DirMax
else
    Edge = false
    Dir = 000
```

Project Description

The students are required to implement a real-time Sobel Edge Detection (SED) on an FPGA module. Traditionally, SED is done on still images. This project may use a Camera connected to a PC to continuously capture images, transfer an image or part of an image to the FPGA, SED is run in real-time on an FPGA, the resulting image is uploaded back to the PC, and displayed the monitor. This motivates student about computer vision and digital image processing.

Appendix:

Conversion of a RGB Image to Grayscale

In order to perform edge detection, we would need to convert the RGB image from the camera to a grayscale image. In a grayscale image, the green component is required to be higher in intensity. We first tried to perform the conversion by scaling down the RGB image:

$$\text{Grayscale} = 0.3 * \text{red} + 0.6 * \text{green} + 0.1 * \text{blue}$$

The closest that represents this to implement on hardware efficiently is as follows:

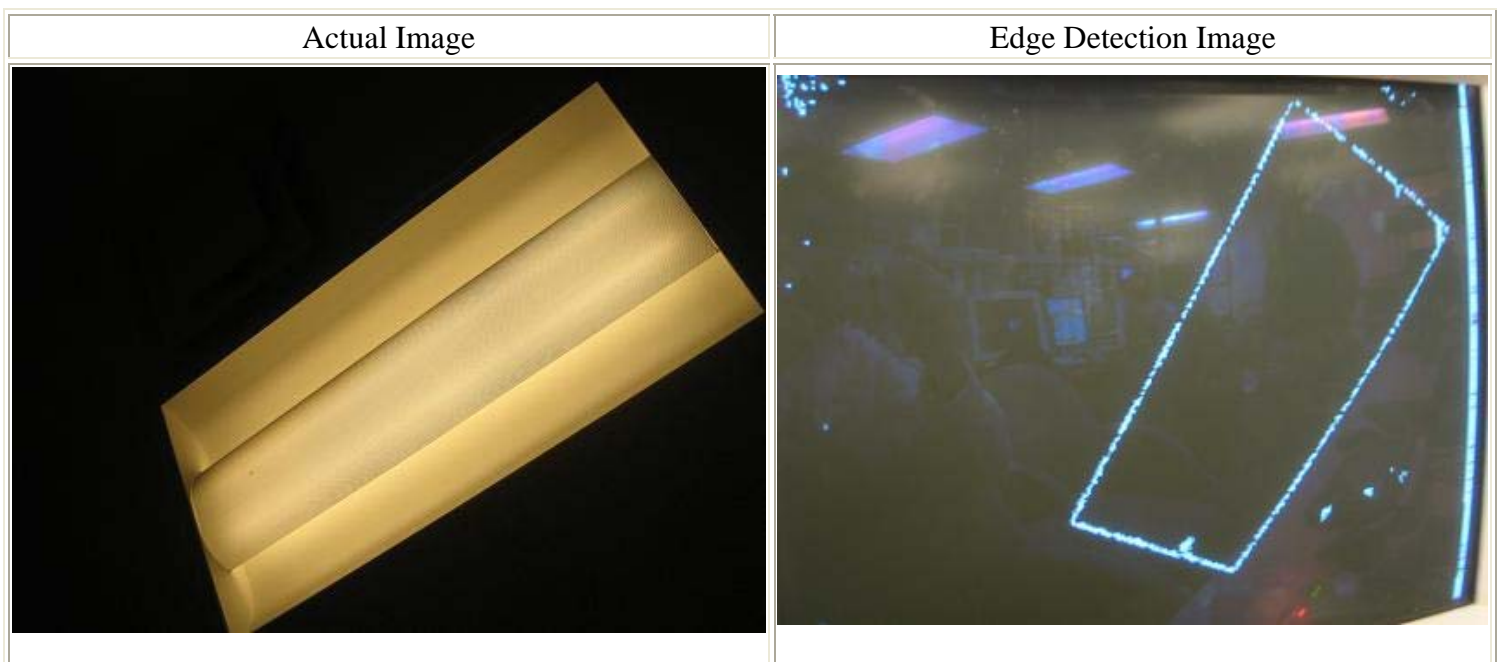
$$\text{Grayscale} = \text{red} \gg 2 + \text{green} \gg 1 + \text{blue} \gg 2$$

However, the resulting image did not achieve enough contrast for edge detection. After several refinements, it is discovered that scaling up the green give the best result for the DE2 camera.

$$\text{grayscale} = \text{red} + \text{green} \ll 1 + \text{blue}$$

Results

The following images show the result of the threshold grayscale image before edge detection, and the result after edge detection. The following image shows the camera pointing towards to the ceiling to edge detect the light. We can observe that the clear outline of the light is detected. It was unable to detect the details inside the light. This is due to the fact that edge detection is performed on a grayscale scale. In grayscale, it is difficult to threshold the difference between areas with high brightness.



COE 306: Automatic Elevator

The project is based on designing & building an FPGA Module to Control the operation of an 8-floor elevator which we denote by E.

Elevator State Vector E

The state of the elevator E is defined by an 8-bit state vector denoted as $S[i]$, where $S[i]=1$ if the elevator is currently at the i th floor of the building. We always must have some component of $s[]$ being 1 and no more than one component can be 1 at any given time.

Elevator Desired Button Vector D

We also have inside the elevator room 8 binary entries denoted by $D[i]$ to allow the users to select the desired floor. Entries $D[i]=1$ if one user presses the button $D[i]$ at any time, $D[0]$ is the ground floor and $d[7]$ is the seventh floor. For example $D[]=00001110$ if there are users to go to floor 4, 5, and 6. Note the following:

- If $D[i]=1$ and now $S[i]=1$, then we can clear $D[i]=0$ because E is now at the i th floor. In other words when E state changes to a new state (like $S[i-1]=0$ and $s[i]=1$) then we must check $D[i]$ and clear it. We assume the service of the i th floor is satisfied when E passes by the i th floor.
- It clear that $D[]$ must be considered as input to decide how to move E up or down. Another important input is the E state which is $S[i]$. Therefore the next state of E^+ must be function of $E^+(D[],E[])$. For example $D[]=00001110$ and $E[0]=1$ (meaning $E[]=10000000$), then E must move up to serve floors 4, 5, and 6. Therefore $E^+(1)=1$ to indicate the E moves to 1st floor and it is no more at ground floor $E[0]=0$.

Simple Solution

Note that the next state E^+ must be decided based upon the i th position satisfying $E[i]=1$ and vector $D[]$. A generic approach would be to move up if “the number of 1s to the right of the i th position in $D[]$ ” is greater than “the number of 1s to the left of the i th position in $D[]$ ”. Lets’ use the following notation:

- Variable Left is “the number of 1s to the left of the i th position in $D[]$ ”, Left can be 0 if $S[0]=1$ where E is at the ground Floor.
- Variable Right is “the number of 1s to the right of the i th position in $D[]$ ”, Right can be 0 if $S[7]=1$ where E is at the 7th Floor.
- The above suggest that to the change the state of E we need to compute variables Left and Right. And compare them. If $Left \leq Right$ then E must move up (more cabin users in upper floors) to serve the largest number of cabin users. On the other hand, If $Left > Right$ then E must move down to serve the largest number of cabin users in lower floors.

Engineered Solutions

A more elaborated and optimized Service Approach is decide (move up or down) the next state E^+ based upon (Left , Right, and $R[i]$), where $R[]$ is a 8-component request vector. $R[i]=1$ corresponds to the elevator request button which is available (outside the elevator) at the i th floor. Each component $R[i]$ can be in one of 3 states:

- 0 (no request),
- +1 if there is a request to move to a floor which is above the i th floor, and
- -1 if there is a request to move to a floor which is below the i th floor.

Therefore, the next state E^+ based be function of (E, Left , Right, $R[]$), i.e E^+ should be function of current state E, the number of cabin users asking to go to the lower floors (Left) and those asking to go to upper floors (Right), and the request vector E for the waiting users at the various floors.

Miniproject Objective Functions

Design the automatic elevator controller by deciding how to update the state of E. In other words the students are requested to determine the relationship that allow what should be E^+ based on function of (E, Left , Right, $R[]$) to satisfy some optimization criteria like:

- Have a very simpler criteria for evaluating $E^+ = (E, \text{Left} , \text{Right}, R[])$
- An optimized criteria like evaluating $E^+ = (E, \text{Left} , \text{Right}, R[])$ to serve the largest number of users.
- An optimized criteria like evaluating $E^+ = (E, \text{Left} , \text{Right}, R[])$ to make the least number of travels to serve all the request from inside the cabin and from the various floors.
- An optimized criteria of your choice.

Appendix C. PROJECTS ON HARD/SOFT INTEGRATION

COE 306: Infrared-Based Home Security System

Thanks for Mr. Mohammad Felemban for compiling this project.

Reference Paper:

<http://ieeexplore.ieee.org/search/searchresult.jsp?newsearch=true&queryText=Design+and+Implementation+of+Pyroelectric+Infrared&x=15&y=17>

Summary:

The objective of this project is to design home security system. The system composed of low-cost motion detection sensors. Moreover, the system will notify the home's owner of any unusual movement with different communication means.

Description:

The idea is to design a home security system using low-cost components that is able to notify remote home's owners. The system is composed of Passive Infrared (PIR) sensors built around a microcontroller. PIR sensors passively accept incoming infrared radiation from surrounding objects. If PIR sensors detected unusual movement in front of it, it will generate pulses to the microcontroller. According to the pulse received, the microcontroller will warn the home's owner by sending a text message using GSM module. On the other hand, the security system remains in idle position and performs nothing if no movement is detected. The block diagram of the system is shown in Figure 1.

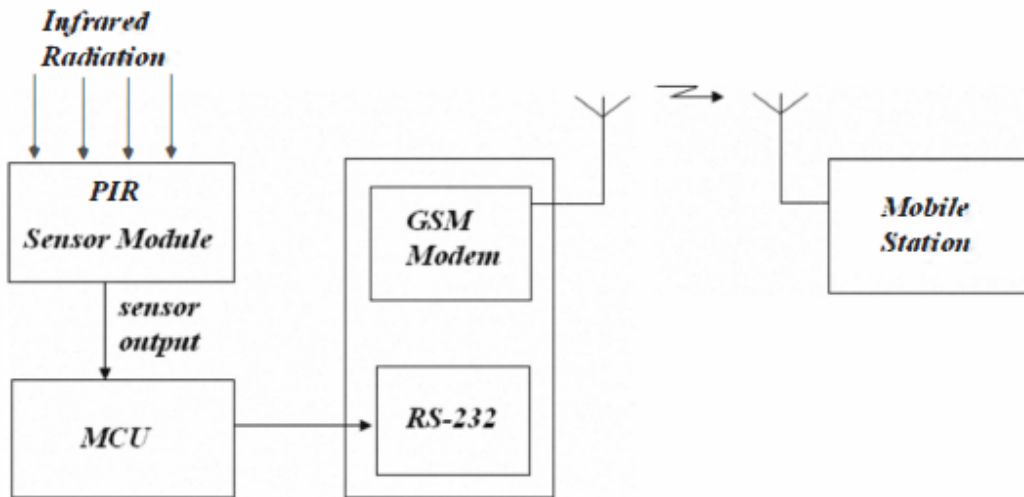


Figure 1: System Block Diagram

PIR sensors are made of pyroelectric sensors that generate electric signals in response to a change in the current thermal radiation. Commercial PIR sensors include two IR-sensitive elements with opposite polarization. When the sensor is idle, both sides detect the same amount of IR. When a warm object passed by, one side detects it first causing a positive differential change between the two halves. As the object leaves the detected area, the other side detects the movement causing a negative differential change. A chip called Micro Power PIR Motion Detector IC takes the output of the sensors, processes it, and generates digital output pulse from the analog sensor. The schematic output waveform is shown in Figure 2.

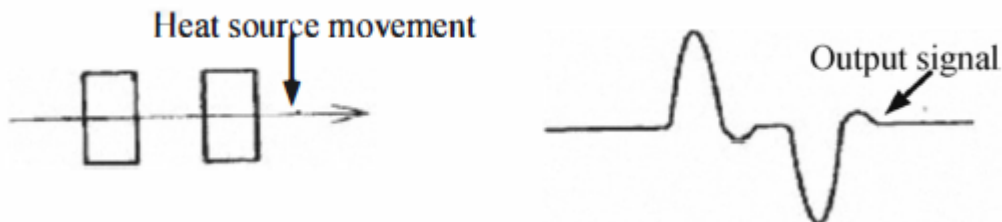


Figure 2: PIR sensor output waveform

The system can be divided into three segments:

- 1- Sensor and Signal processing segment

The segment is shown in Figure 3, and is composed of five parts;

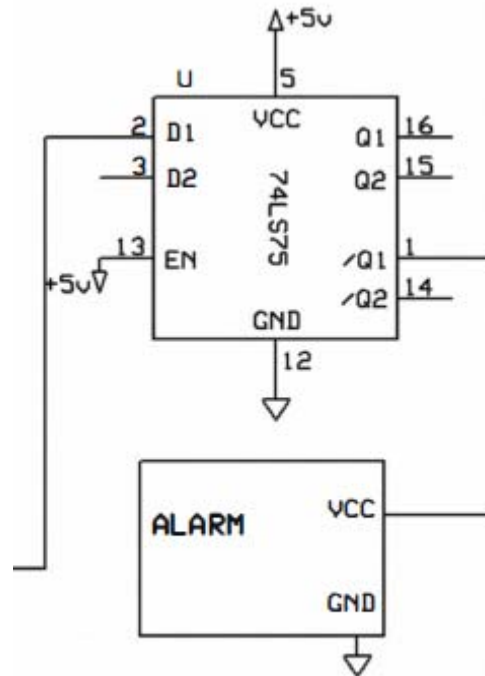


Figure 4

- a. 74LS75: D-latch IC. The input voltage level on D1 is kept unchanged on Q1 and inverted on Q2. Inverted output represents the voltage level set by MCU keeping the alarm on even when MCU is idle.
- b. Alarm: the alarm has two pins, VCC and GND. VCC pin is connected to the /Q1 of the latch.

3- GSM Modem interfacing segment:

This segment is shown in Figure 5, and consists of three parts

- a. MAX232: this IC is responsible for increasing the voltage swing at the output. It takes 0V and +5V inputs and makes it -12 V and +12 V output voltages. This increasing is required for serial communications. Two capacitors are used for this particular reason as shown in the Figure.
- b. DB9 male connector: since the GSM module uses a female DB9 connector, pin 14 and 13 of MAX232 are connected to pin 2 and 3 of DB9 respectively. Pin 5 is grounded.
- c. GSM module is connected to the through a DB9 female connector to the interfacing circuit.

The circuit is operated as the following. External interruption of MCU is disabled by software. The system is turned on when the switch is closed. LM35 senses the temperature and converts the analog voltage to binary value and sends to the MCU. The MCU monitors the temperature and light the LED when the temperature is equal or greater than the critical temperature. After

the LED is on, the MCU waits for a pre-defined time until the movement is settled. After the time is elapsed, the system is activated. After the activation, if there is more movement in the detection region of the PIR, it outputs a pulse which is taken by the MCU. MCU then waits for another pre-defined time and checks for the signal again to avoid false triggering. During the first cycle of the D-latch, Q1 is High. Hence, $\overline{Q1}$ is low. If the signal is still present, MCU will make the output signal low during the second cycle. And hence $\overline{Q1}$ will be high and the alarm rings. In the same time, the MCU will activate MAX232 IC. Then MCU starts to send AT commands to the GSM module serially. The modem receives the command and set-up a call session to the pre-defined number of the home's owner. The call gets disconnected when the call time-up. The MCU then enables the external interruption and goes to sleep. The flowchart of the circuit operation is showed in Figure 6.

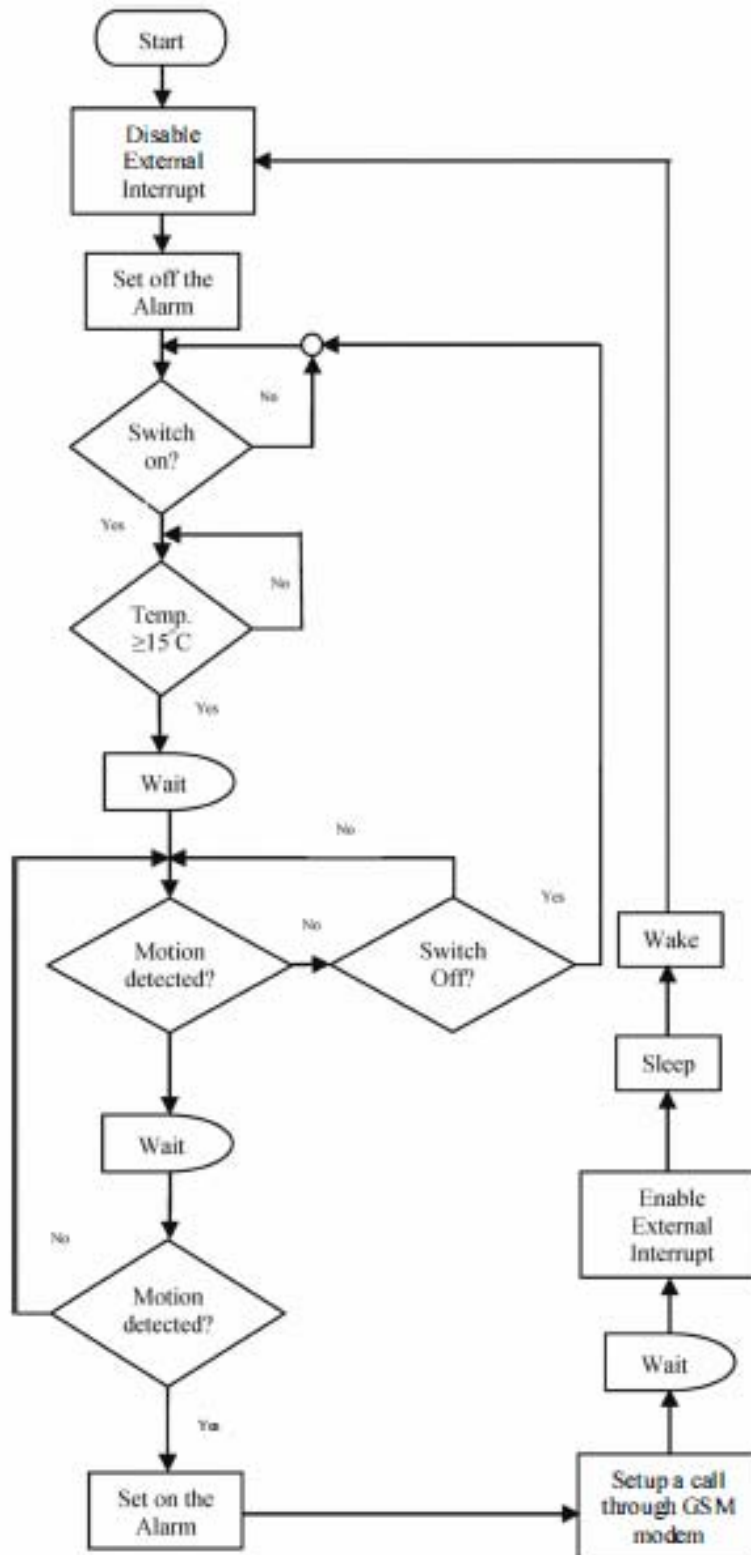


Figure 5

Thanks for Mr. Mohammad Felemban for compiling this project.

Reference Paper:

<http://ieeexplore.ieee.org/search/searchresult.jsp?newsearch=true&queryText=automated+voice+navigation&x=36&y=18>

Summary:

The objective of this project is to encourage the use of off-the-shelf speech recognition circuits along with a microcontroller to implement automated systems that support elderly and physically challenged people. Available speech recognition circuits recognize up to 40 words and convert them into digital signals. Using proper interfacing circuits to microcontrollers, such speech recognition circuits are used to control electrical devices using voice commands.

Motivation:

Elderly and physically challenged people mostly depend on others for their livings. Most of the time, they use the help of others to do simple yet substantial tasks, e.g., moving wheelchair around the house, turning book's pages, etc. Studies have shown that until quite recently disabled people were socially isolated if they didn't find people to take care of them. Having known these facts, the aim is to design user-friendly voice-enabled systems to help them.

Description:

I. Voice Recognition Interface

Speech recognition circuit (SR-07) is a programmable circuit that can be trained by the words or vocal utterances. Once the circuit is trained, it recognizes the words and converts them into digital signals. The heart of the circuit is the HM2007 speech recognition IC which recognizes up to 20 2-minute words or 40 1-minute words. The circuit diagram is shown in figure 1. SR-07 circuits are interfaced to microcontrollers using speech recognition interface circuit (SRI-03). SRI-03 incorporates ten logic high outputs corresponding to the ten train words by SR-07. Using this interface easily allows activating and deactivating 10 electrical devices or circuits. For example, trained word number 1 on SR-07 activates output 1 in SRI-03. Trained number 2 activated output 2. And so on. The interface circuit is shown in Figure 2.

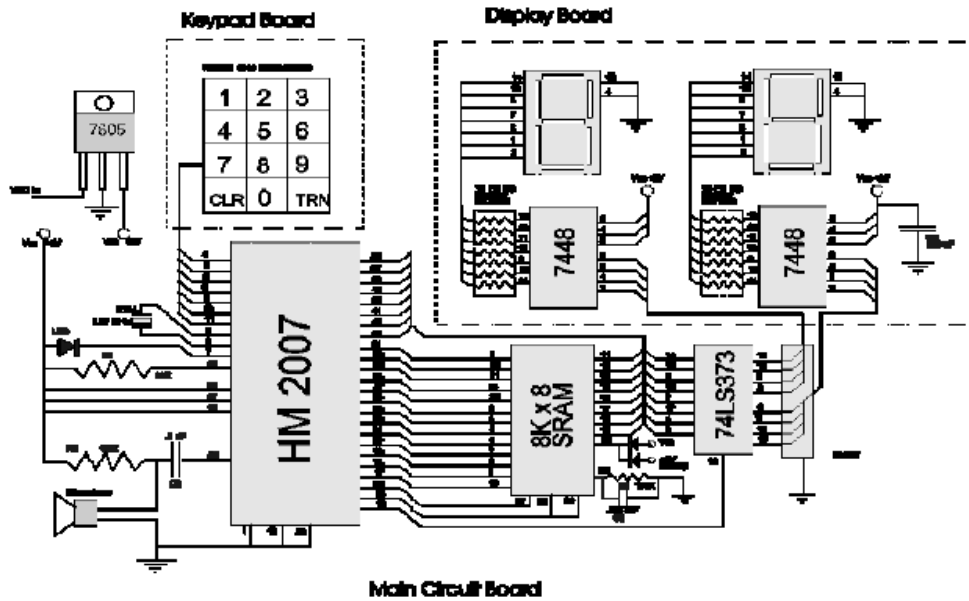


Figure 6

Interface Circuit

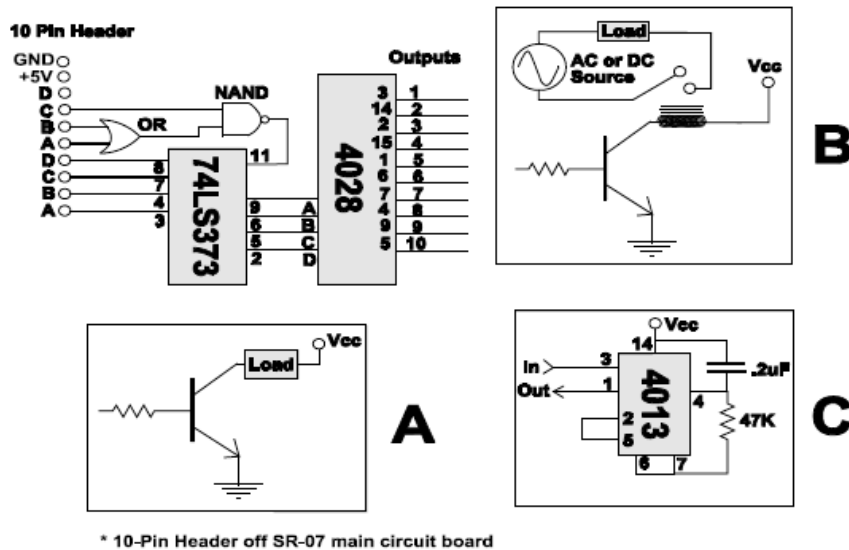


Figure 7

II. Words Training

There are three styles of speech: isolated, connected, and continues.

- Isolated: words are spoken separately with a pause between each word
- Connected: in the middle way between isolated and continues speech. Multiple words can be spoken.
- Continuous: resembles the natural conversational speech which contains everyday sentences. This very difficult to recognize as the words tends to merge together

SR-07 speech recognition circuit is able to recognize the isolated and connected words. However, the words training process depends on the speaker. SR-07 support two ways: speaker dependent and speaker independent.

Speaker dependent systems are trained by the individual who will be using the system. These systems are capable of achieving a high command count and better than 95% accuracy for word recognition. The drawback to this approach is that the system only responds accurately only to the individual who trained the system. This is the most common approach employed in software for personal computers. Speaker independent is a system trained to respond to a word regardless of who speaks. Therefore the system must respond to a large variety of speech patterns, inflections and enunciation's of the target word. The command word count is usually lower than the speaker dependent however high accuracy can still be maintain within processing limits. Industrial applications more often require speaker independent voice recognition systems.

COE 306: Voice based Home Navigation System for the Physically Challenged

Thanks for Mr. A. Arafeh for compiling this project.

Reference Paper: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5745885>

Project's Description:

The elderly and physically challenged people often require some assistance to perform their daily routine. The most common image of disability is the people in wheelchairs who require external aid to move around. The aim of the project is to develop a simple Home Navigation System (HNS) which is based on wheelchair with voice commands capability. The wheelchair should automatically navigate from its current position to the destination as per predefined route and voice command.

Project's Modules:

The Home Navigation System (HNS) should include the following modules:

- 1- Voice Capture Module: This module is responsible of capturing voice commands produced by the person using the wheelchair and converting them into digital signals.
- 2- Line Follower Module: This module helps the wheelchair to automatically navigate inside house by following the black line on the floor of the house.
- 3- IR -TSOP Module: This module communicates with the IR transmitters installed at each room door. IR transmitters will be sending a specific data identify the room that the wheelchair has reached its entrance.
- 4- Motor Control Module: This module works on the commands received from Voice recognition module. Stepper motors can be used to rotate through various directions. Speed and direction of the motors are controlled by the ARM processor.
- 5- Personal Security Module: This is an emergency module that has GPRS modem for sending SMS emergency messages. The user can avail this module by voice command if they feel uncomfortable or requires immediate attention.
- 6- Obstacle Detection Module: This module uses Ultrasound sensors to detect an obstacle in the proximity.

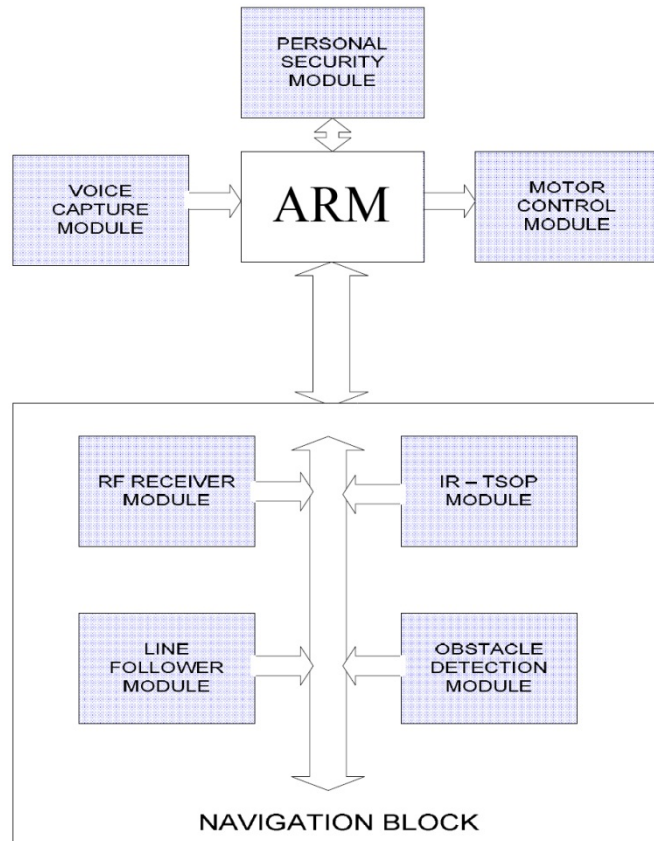


Figure 8: Block diagram of the Home Navigation System

Application Program

The application program should satisfy the following simple operations:

- At the beginning the voice capture module awaits for voice commands. Specific voice commands should be preloaded to identify different places (i.e., rooms), or for general directions like left, right, front, back etc.
- In general directions commands, the microcontroller drives the motor control module according to the direction voice command.
- When the voice for a specific room is detected then a simple line detection procedure is performed. The wheelchair moves straight until a line or an obstacle is detected as shown in Figure2. If obstacle is detected then it takes left and moves in straight line until it detects the line.
- When the line is detected, the wheelchair follows the line till it receives any signal from IR TSOP. The received data from TSOP is compared with the predetermined data bits stored in the system, which denote the destination room. If destination room has not been reached the wheelchair continues to follow the line.

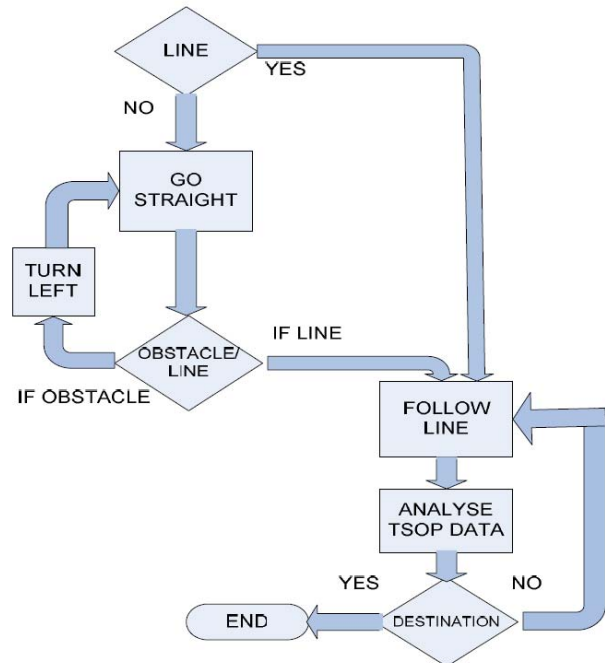


Figure 9: Application Program Flowchart

Testing and Experiments

Figure 3 shows the structure of a sample room which can be used to experiment the system. It consists of four rooms namely the kitchen, living room, bed room and drawing room. Each room has a door and a TSOP is attached to each door. The route is identified as bold black line with junctions A, B, C and D. The wheel chair can move along the line similar to 'line follower' robot. This line must be drawn inside the house for the wheelchair to move along.

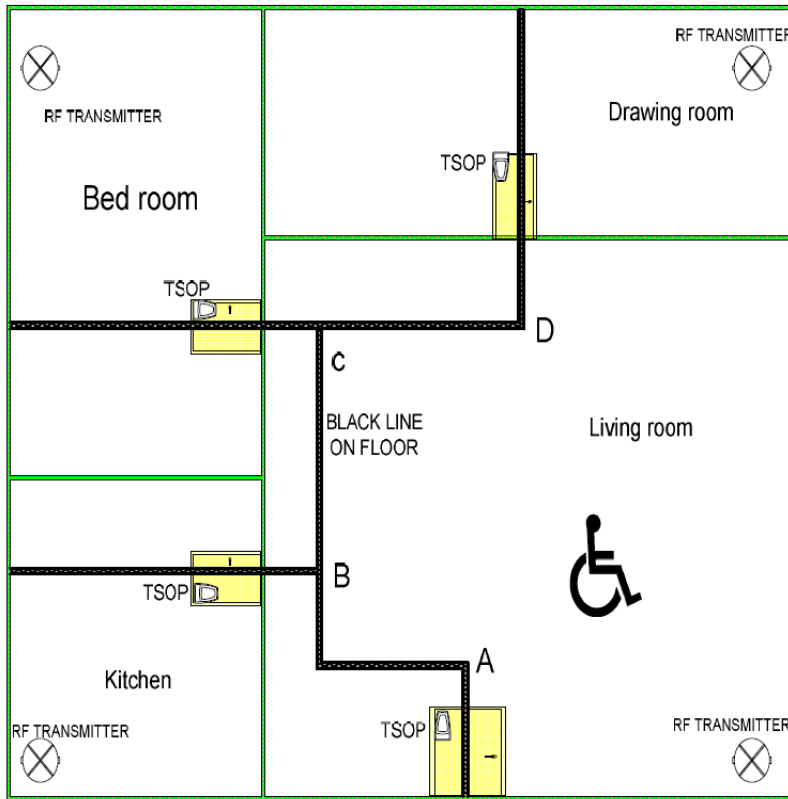


Figure 10: Simple Home Structure

COE 306: Unicursal-1 Gesture Interface for TV Remotes with Touch-pads

Thanks for Mr. A. Arafeh for compiling this project.

Reference Paper: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5722941>

Project's Description:

TV sets are becoming more and more equipped with Network access capability, many kinds of network services (Internet browsing, Video on demand...etc.) are now accessible via TV screens. Since each of these new services requires new input commands, conventional remote controllers are falling short in providing smooth and relaxed experience. The aim of the project is to develop and extend a gesture-based interface that can be used as a remote control to access hierarchical menus in network capable TV sets.

Project's Details:

Unicursal gesture interface is based on thumb gesture preformed on touchpad. The interface should be able to detect shapes drawn by a thumb in single move (i.e. circle or square) and the orientation of the shape in the touchpad. An example of such shapes is shown in Figure 1, the type of the shape and the orientation can be used to identify different commands to the TV set.

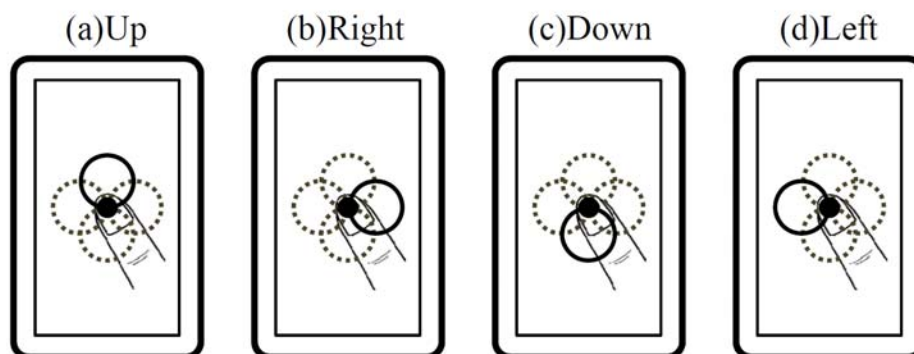


Figure 11: Gesture orientation of Unicursal figure (circle).
Black point shows start position and end position of the figure

Different shapes can be implemented using the same notion described above. Figure 2 shows four types of unicursal shapes.

¹ In mathematics it denotes a closed curve which can be drawn in a single movement.

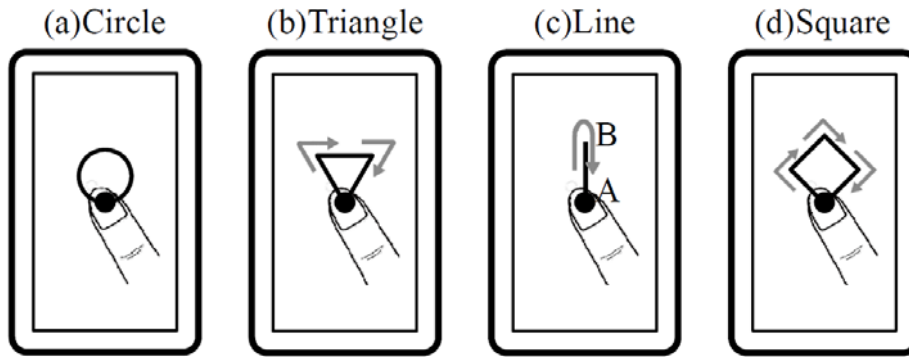


Figure 12: Examples of Unicursal Shapes

Implementation of Gesture Interface

The implementation of gesture interface depends on developing an application program that detects gesture start and end points. Hardware implementation requires an ARM processor with touchpad around 3.5inch diagonal and 480- by-320-pixel resolution as shown in block diagram in Figure 3.

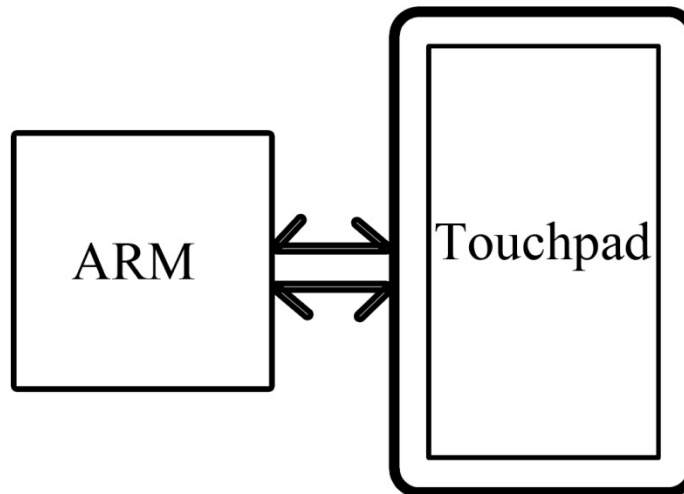


Figure 13: Interfacing with ARM Processor

One way of implementing a gesture recognition application is to use detection zone, which is the circular or square zone shown in Figure 4. The center of the detection zone is the point at which the user's thumb first touches the touchpad. A gesture starts when the forefinger leaves the detection zone and ends when the forefinger returns to the detection zone. The point at which the finger leaves the zone and return determines the orientation of the shape and its type.

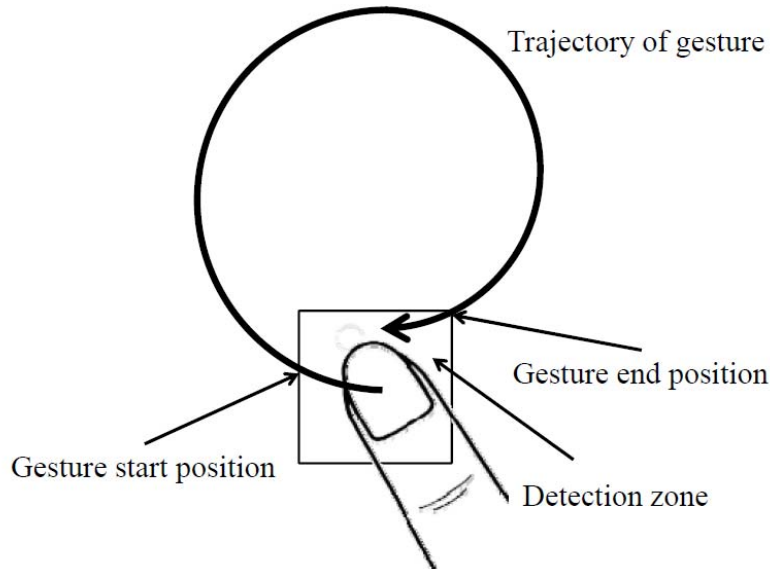


Figure 14: Detection Zone with start and end positions

Application Program

The following steps shows an example of implementing a gesture recognition application for four shapes: Left line, Right line, Left circle and Right circle. Figure 5 shows the application of those four shapes in EPG menu of TV set.

- First, the recognition application identifies the detection zone around the position where user's thumb first touches the touchpad.
- Second, the application detects that user's thumb return to the detection zone after user's thumb leave the detection zone.
- Third, the application detects orientations. If x-coordinate of the point where thumb leaves detection zone is left side from x-coordinate of the center of the detection zone, orientation is left. As wellfor x-coordinate and y-coordinates of the other three orientations.
- Finally, the application recognizes the kinds of unicursal shape. The application detects the y-coordinates of leaving and returning to detection zone. If the difference between theses coordinates is above a threshold, the application recognize circle. If the difference is under a threshold, the application recognizes the unicursal line.

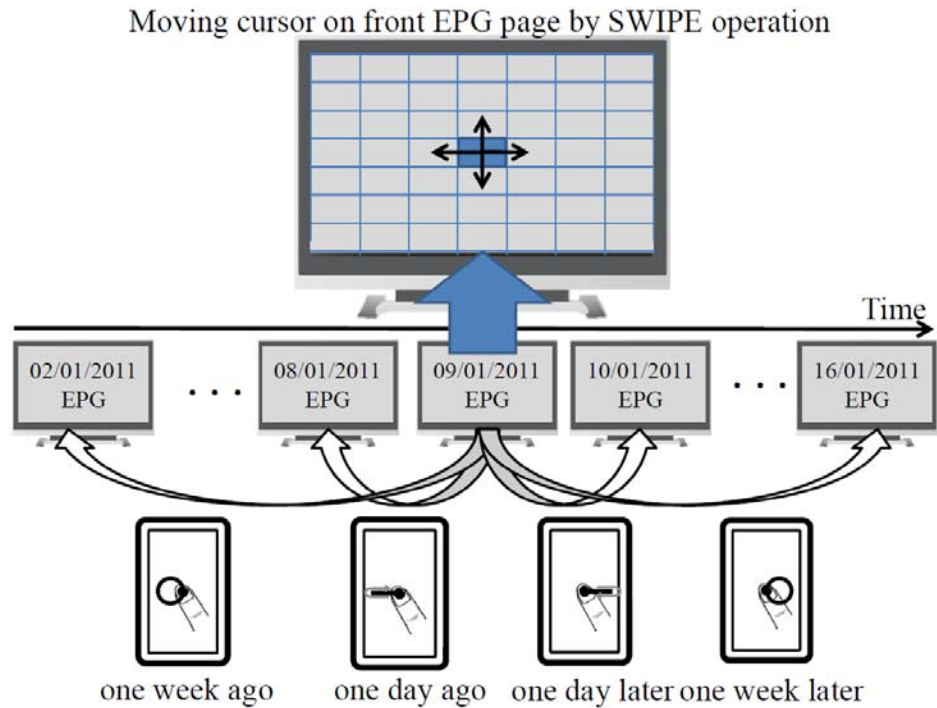


Figure 15: EPG control example

Testing and Experiments

Experiments can be performed to measure the ratio of detection of each shape (i.e., command). Additional uni-cursal commands can be tested. Testing should also be conducted to account for cases when thumb fails to return back to detection zone and devise different techniques to overcome different detection failures.

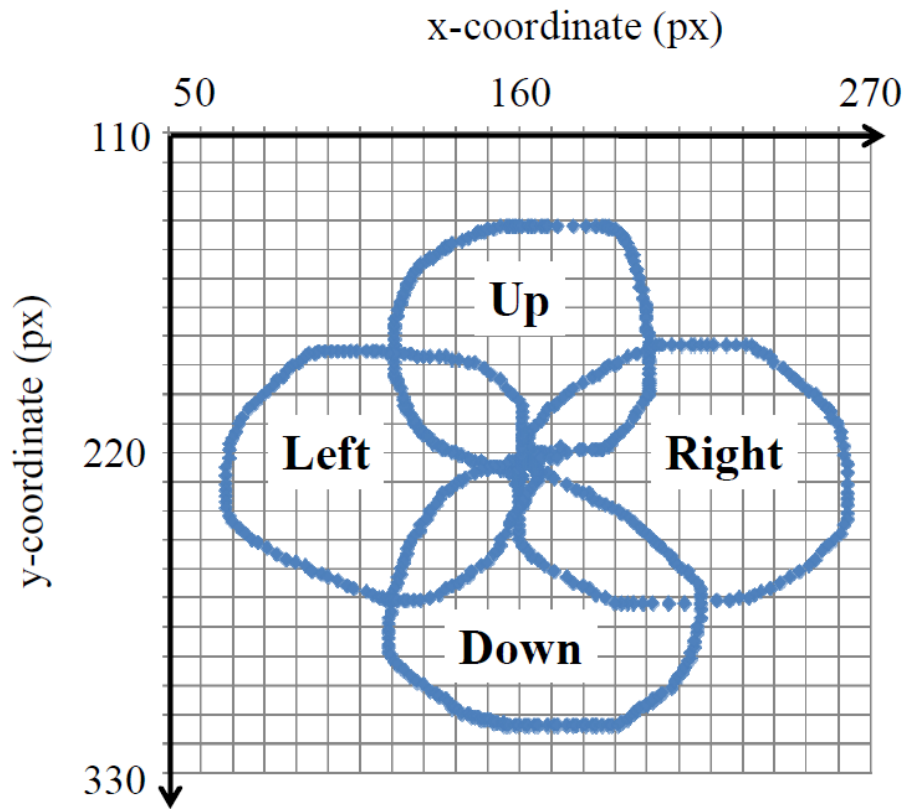


Figure 16: Example of traces generated on the touchpad by drawing unicursal circle shape

COE 306: GSM-based Automobile Anti-Theft System

Thanks for Mr. A. Al-Suwaiyan for compiling this project.

Reference Paper: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4777403>

Project Summary

You are to implement a GSM-based automobile remote alarm system. The system will be using TC35 GSM module, a Microcontroller, a vibration sensor and a speed sensor. The purpose of the TC35 GSM module is enable short message service (SMS) alarm. Two levels of anti-theft are to be implemented: the first level utilizes the vibration sensor to detect theft trials by, for example, opening of doors illegally. In case a vibration signal is detected, light and sound alarms are activated, oil supply is cut, and an SMS is sent to the owner. In case no cancellation request is made by the owner, dial the owner's telephone. If the thief manages to get away with this level, second anti-theft is activated, in which speed sensor measures the speed of the vehicle and when it reaches 15 m/s, it again sounds and light alarm, cut off oil, sends an SMS to notify the owner at the same time. Similarly, in case no cancellation request is made by the owner, dial the owner's telephone. In case the thief manages to bypass both anti-theft proofs, on-board communications device signals can be base-station searched through GSM mobile communications network, which can provide some clue for retrieving vehicle.

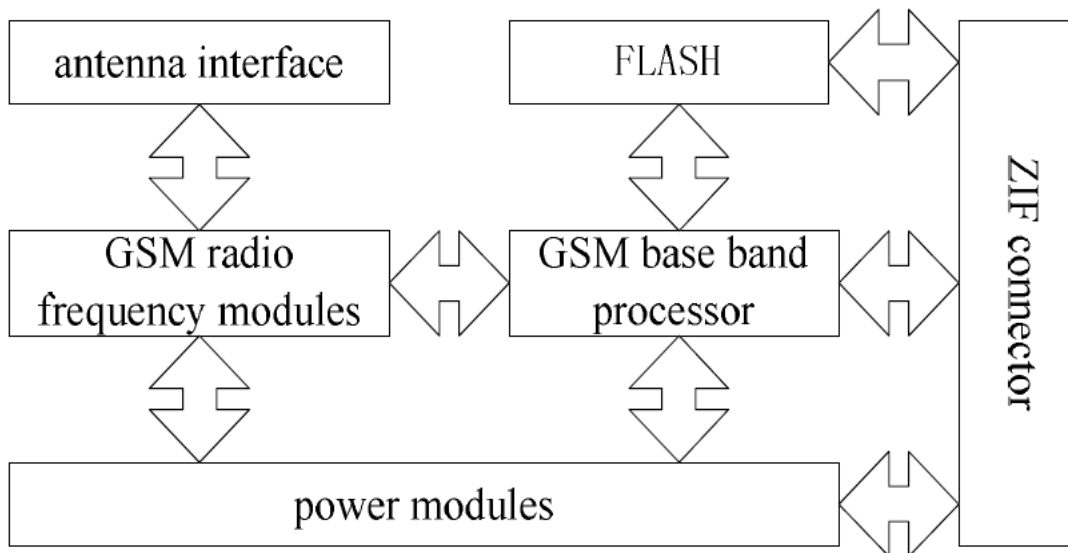
System Components

Number of vehicle is increasing rapidly in the world, and this implies the chance that a vehicle might be stolen is increased as well. An effective anti-theft system is a crucial element of modern vehicles.

In this project, you will be implementing an anti-theft system which will be integrated with the GSM network for owner alert purposes. The system is composed of the following components:

1. **TC35 GSM network module:**

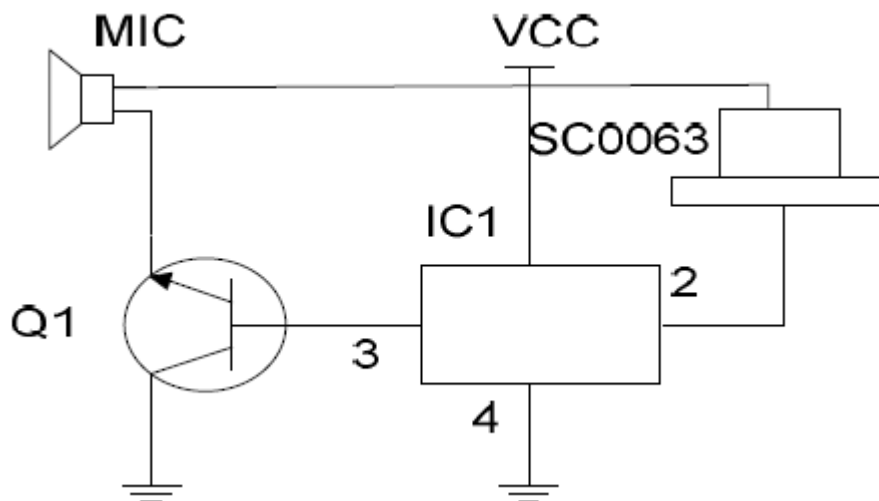
This is a major component of the system and it enables integration with GSM network. Here is a list of facts and features of this component. It is a wireless module produced by Siemens. It can realize data and voice transmission quickly and safely, such as short message services SMS and fax. It can work in GSM 900kHz and 1800kHz two bands. RS232 data port tally with ETSI standard GSM0707 and GSM0705. It's easy to be upgraded to GPRS module. The module which set RF Circuits and base-band together provides users with the standard AT command interface can make fast, reliable, secure transmission of data, voice, fax and SMS. It's easy for users to develop and design. TC35 module is composed mainly by six parts: GSM base band processor, GSM radio frequency modules, power modules (ASIC), flash memory, ZIF connector, antenna interface. The functional block diagram of this component is shown below.



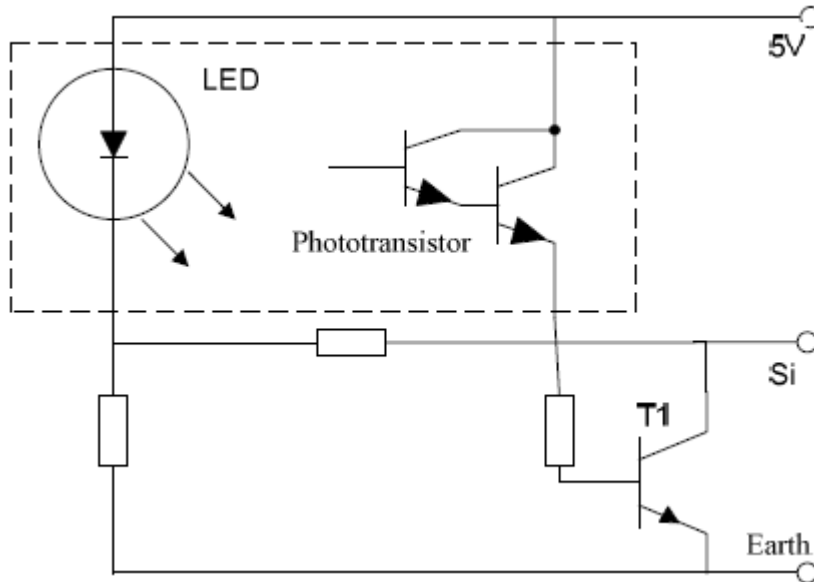
As the core of TC35, base band processor with the responsibility of disposing voice and data signal in the terminal of GSM, and covers all of the analog and digital functions in cellular radio equipment. It supports FR, HR and EFR voice channel coding with no additional hardware circuitry premise.

2. **Vibration sensor:**

This component is used to detect any vibration in the car, for example, breaking one of the windows, severe crash, and breaking one of the doors. All these events cause vibration, which are detected by the vibration sensor. Vibration sensors are put on four doors, front cover and trunk. SC0063 displacement vibration sensor is to be used. It isn't restricted by the direction of vibrations, and its perceived sensitivity are the same whatever at the upper and lower, left and right, front and rear. The sensors are between the IC trigger-end and power of general alarm sensors. It will trigger alarms by the degree of vibration detected. The circuit is shown below.



3. **Speed sensor:** This will detect if the speed of the vehicle exceeds a predefined threshold. This sensor will be used in the second level protection. Photoelectric speed sensor configuration is used in the speed sensor, which is constituted of light-emitting diodes, Phototransistor and light shade board. Its working principle is shown below.



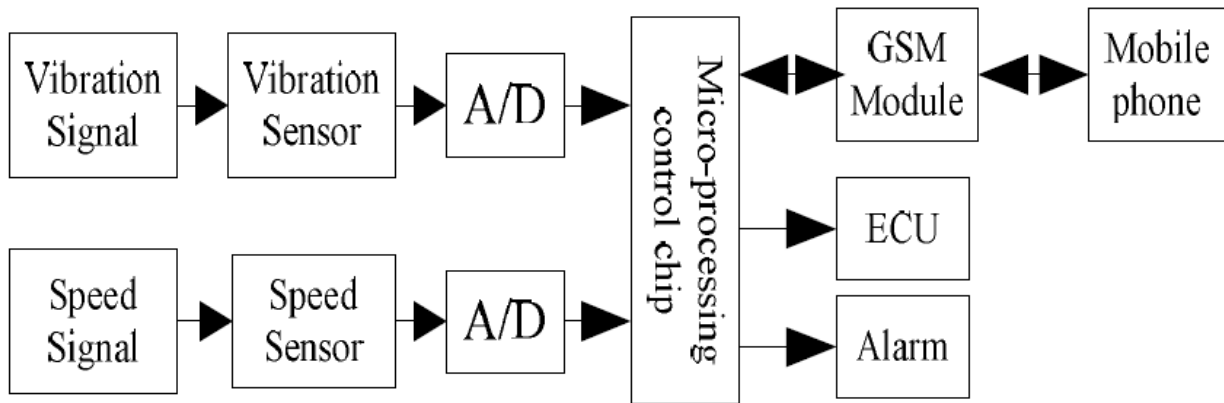
When the light shading board can not trip the speed of light, light of LED reach the Phototransistor, current pass the collector of Phototransistor, then the tube turn on, transistor VT1 also turn on, so the Si terminal output 5V voltage. Pulse frequency is decided by the speed.

4. **AT89C52 Controller:**

This is the beating heart of the system which integrates and operates all parts. AT89C52 of ATMEL Corporation which is a low-power, high-performance single-chip with 8K bytes internal Flash Memory is selected. Advanced high-density non-volatile memory technology of ATMEL is used in the chip. It's fully compatible with instruction set and output pins of 51 series. There are 256 bytes RAM, 32 programmable I/O Line, 3 x 16-bit timing counters, 6 Two-stage interrupt sources, and a full-duplex serial port.

5. **A/D Converters:** Two Analog-to-Digital converters which are required to encode the vibration and speed analog signals into digital signals. Details of this component will be discussed in the interfacing part of this document.

The following figure shows a rough sketch of these components and interconnections among them.

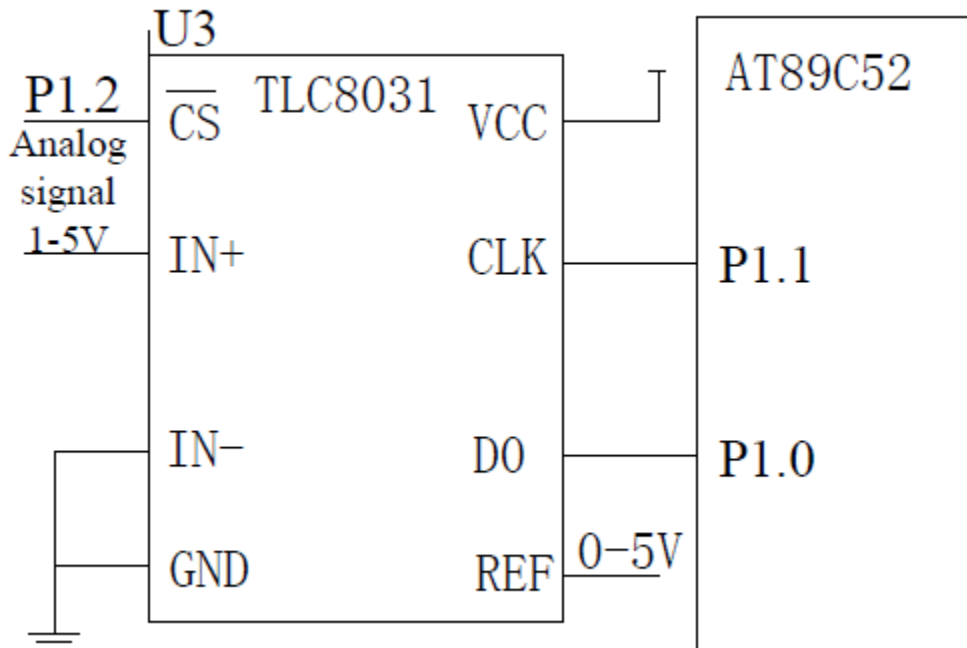


Interfacing

You are to implement two two interfaces:

1. A/D interface:

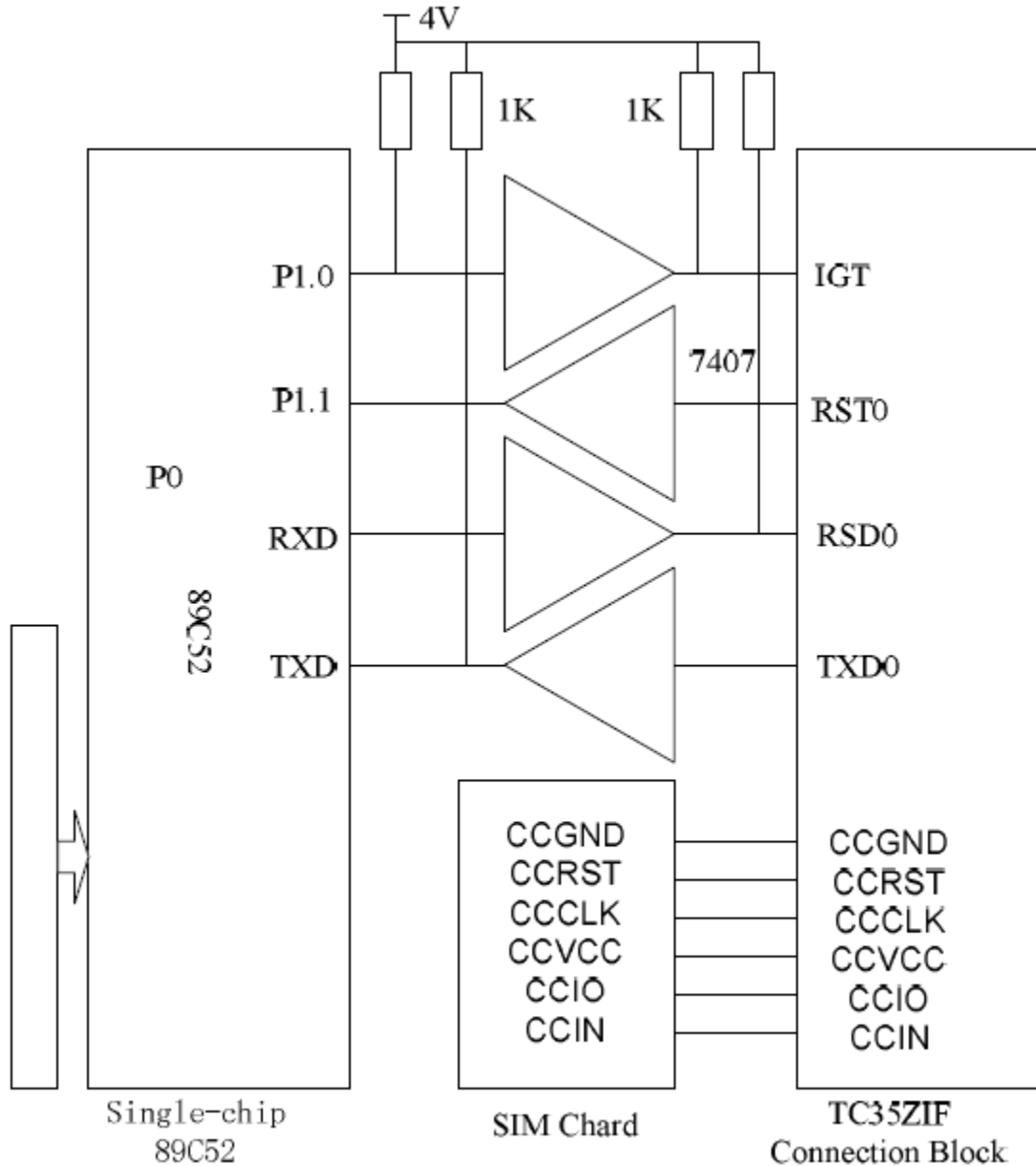
We are going to use the 8-pin package TLC0831 Analog-to-Digital converter. The chip supports single channel analog input, and the converted digital output is stored in an 8-bit shift register. Digital signal is calculated and stored in the shift register, where the least significant bit, D0, is one of the package pin. The following figure illustrates the interfacing between the microcontroller and A/D converter.



2. GSM interface:

TC35's start, reset and shut down is controlled by P1.0 and P1.1 of the single-chip. Data

communications with TC35 through the serial port, by communication rate of 9600Kbps, using 8-bit asynchronous communication: a start bit, 8 data bits, a stop bit. The essential part of the interface between single-chip and TC35 is shown in the following figure.



The following facts should be considered:

- TTL logic-positive of TC35 module's input and output is not +5 V, but +2.9 V. Therefore, input level must have a logic conversion by adding pull-up resistor to the output of the open-collector buffer 7407.
- After power the system, in order to make TC35 working, we must apply a low pulse for more than 100ms delay to IGT. The duration of level decreased is no more than 1ms. After the start, IGT should remain high (3.3V). Supply voltage of TC35 cannot be lower than 3.3V when IGT drive; otherwise TC35 cannot be activated.

- c. CCIN pins of ZIF connector blocks are used to detect whether the SIM card is inserted. If connect right, CCIN pin output high, or low.

Software Design

You are to use C51 for application development. Major control algorithm is given below. Your task is to convert this pseudo code into C51 to control the AT89 controller.

1. Initialize TC35: refer to interfacing section to check the requirement for initializing TC35.
2. Remain in this step until the user secures the vehicle.
3. If a vibration signal is detected or speed exceeds 15m/s while in secure mode then

Cut off oil support.

Fire the alarms.

Send SMS to the user.

Wait for 2 minutes to give the user a chance to unsecure the car; otherwise, dial the user phone to alert him further.

End if

4. Go to step 2.

COE 306: Controlling a 3 Wheel Drive Omni-directional for Line Following

Thanks for Mr. F. Ougali for compiling this project.

Reference Paper: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6108741>

Summary

As omni-directional vehicles have a very high flexibility in movement. The replacement of the conventional wheel system with omni-wheels can provide the robot with multi-directional and sideways movement perpendicular to the wheel's rotation.

The objectives of this project are as follows:

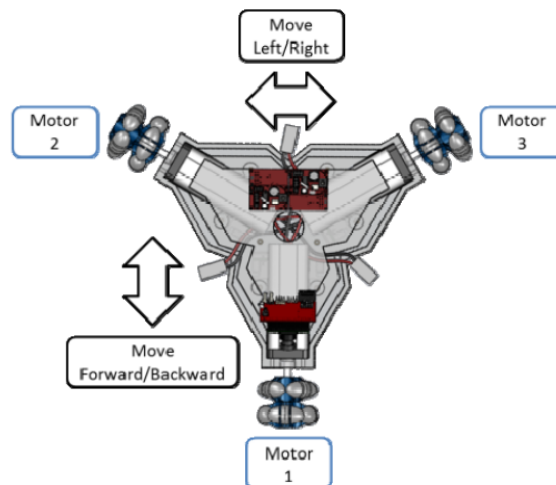
- To design and develop the control for a 3-wheel drive mobile robot by using 32-bit low-power ARM processor.
- To explore the functionality and system architecture of ARM processor.
- To implement a line follower application to the mobile robot by using IR sensors.

Components:

- Microcontroller: NXP ARM7TDMI-S LPC2148.
- Sensors: SUN FX-301 Digital Fiber Sensor (IR sensor).
- Motors: BLH015K-XX* VEXTA DC Brushless Motor.
- Drive Wheels: Omni-wheel.
- Software: WinARM (includes Programmers Notepad Editor and GNU C/C++ compiler).
- The vehicle is supposed to be provided.

Details:

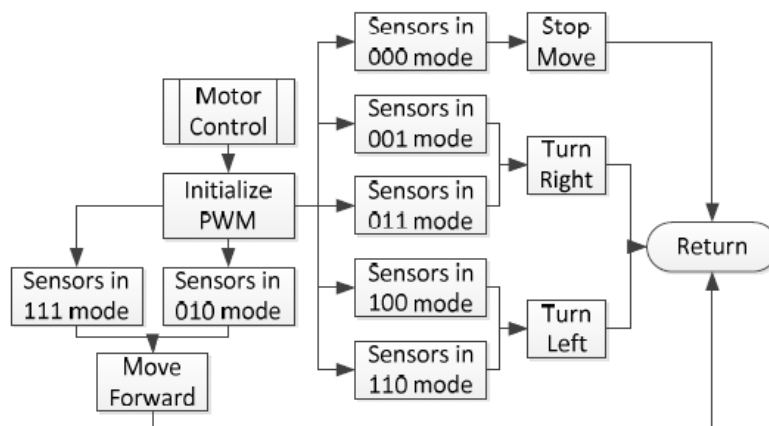
The mobile robot is supposed to keep track of a dark line on the ground, detected by the IR sensors, by continuously adjusting the ratio of speed and rotation direction of different motors (using PWM).



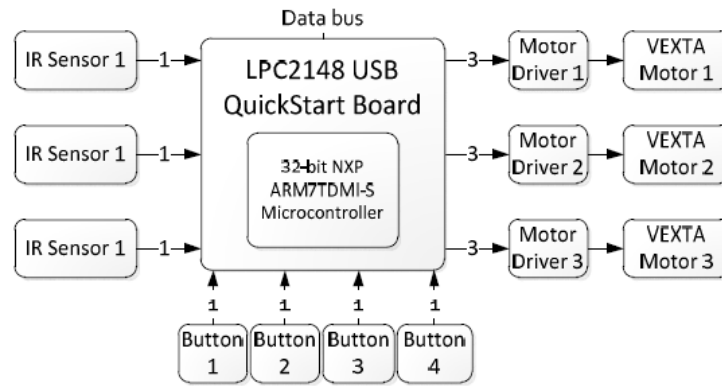
The robot has three motors located in an equilateral triangle form. This structure enables the robot to move forward/backward, to turn left/right, and to rotate around itself.

Robot navigation: To move the robot forward, only motor 1 and motor 2 need to be controlled in counterclockwise/clockwise while the speed of motor 3 should be set to 0. To rotate the robot, all three motors should have the same speed and rotational direction. The speed of the motors can be controlled using PWM outputs.

Line following: Three IR sensors, located in a linear fashion, will be used to keep track of the dark line. Based on continuous readings from these sensors we can decide how to control the motors (move forward, turn left, turn right, or stop). These IR sensors provide a digital output, where logic (1) corresponds to dark colors and logic (0) corresponds to bright colors. Therefore, the IR sensors will produce logic (1) if it detects the dark line we want to follow and logic (0) if it detects the bright surface around the dark line. So, reading (111) means that the robot is directly over the dark line, while reading (100) means that the robot is turning away from the dark line to the right side, and (001) indicates that the robot is heading to the left of the line. The figure below shows how to control motors for all possible scenarios. The reading (010) means that the line we are following is a thin line which is covered only by one sensor, the one in the middle.



Interfacing:



The ARM processor will be used to interface three IR sensors and three motor drivers. Three input pins will be used to read the logic generated by the IR sensors. Nine output pins will be used to control the three motors including running the motor, the speed of the motor using PWM, and the direction of the rotation.

Driving the motors:As for motors drivers, one might design his own driver by using transistor bridges, or use drivers like **A4931** from allegro microsystems.

Source:http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6108741&tag=1