UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA
Faculty of Engineering, Built Environment and Information Technology

Department of Electrical, Electronic and Computer Engineering

Version 18 February 2019

EMK310 practical assignment 2019

Development of a microcontroller-based autonomous robotic vehicle (MARV)

Lecturer: Prof T Hanekom
Email: tania.hanekom@up.ac.za

Broad assignment

Teams of <u>four or less students</u>[1] will design, build and test a microcontroller-based autonomous robotic vehicle (MARV) *from first principles* in three stages, each stage being a practical with specific criteria and deliverables. The objective is for the vehicle to navigate along and complete a specific track by accurately following a specified trajectory, as shown in Figure 1, in the fastest possible time. The tracks are printed on a heavy plastic material. The MARV must stop automatically within 100 mm from the end of the track. The supplied motors must be used. A PIC18F45K22 or PIC18F45K20 must be used as the processor platform. The chassis and other structural components required to construct the MARV must be made from recycled materials or may be 3D printed[2]. No development boards or kits are allowed other than the official Microchip Curiosity development board that you received. Your MARV must fit into a box with dimensions 100 mm x 180 mm x 120 mm (W x L x H) excluding the "chimney" that needs to be mounted on the top of the MARV to trigger the gantry seonsors. The MARV must use one or two rechargeable 9V batteries as a power source.

The MARV will be demonstrated and tested during the Seventh Annual EECE Race Day on Friday 24 May 2019 in the Amphi Theatre on the main campus of the University of Pretoria. **Teams have to race their MARVs to be eligible to gain entry into the examination in this module.** *However, note that for a MARV to be allowed into the race, the team will already have complied with the requirements of the practical of the module. So, even if your MARV becomes a DARV (defying autonomous robotic vehicle) on Race Day, you will have passed the practical component of the module.*

[1] Do not ask to be five or more students in a team because 5>4, and so is 6>4. The answer has always been and will always be "no".
[2] Rules about 3D printing follow later.

Please have your dearest and nearest diarise the date, because they are all invited.

The figure below shows a standard race track. Also visit the [Race Day photo and video archive] to familiarize yourself with the setup.

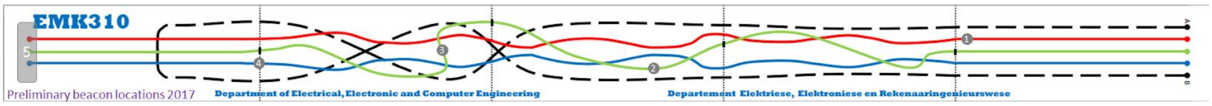


Figure 1. The configuration of The Race showing typical locations of sensors which will earn a MARV tokens when it passes underneath the sensors. The gantry sensors will be placed randomly along the track on Race Day, with each gantry's sensors being located at the same place for all the tracks.

The complete race area consists of 9 tracks (of which only 8 will normally be used), with 5 gantries placed over the tracks. Sensors are located on the gantries and are aligned with specific lines on the track.

a. Project breakdown

The project is divided into three sections, which will be delivered on three dates (three practical demonstration dates).

1. MARV's communications, data transfer and storage, and test protocol (Demonstrate for Practical 1).
2. MARV's sensor system, calibration system, navigation system and touch start system; integration of all subsystems (Demonstrate for Practical 2).
3. MARV's motors and motor control; battery charger and integration of all systems (Demonstrate for Practical 3).
4. Fine-tune MARV for super speed and awesome accuracy (Demonstrate at Race Day 2019).

b. Special rules, recommendations and notes

i. The MARV

Students are encouraged to construct their circuits on veroboard (stripboard), blob board or PC board (as opposed to breadboard) to reduce the weight of the circuit and to ensure robustness. **A drop test from 500 mm above floor level may be performed at any demonstration.** This is crucially important for Race Day where you need to carry your MARV around in an environment where there is a risk for dropping it.

Apart from the electronic circuitry, batteries, motors and wheels of the MARV, and 3D printing filament, no new materials may be used in its construction, i.e. only recycled materials are allowed. An inexpensive source of old toys, old equipment such as printers, etc. is the Sungardens Hospice Shop (18 Twig Street, Corner of January Maselela Drive and Lynnwood Road, Lynnwood Glen). Soft drink tins, ice cream or margarine tubs, container lids, bottle tops, etc. are examples of materials that may be used.

- Recycled chassis and/or other parts of radio controlled vehicles are examples of materials that will *not* be allowed.
- Lego, Meccano or similar construction toys, even if dating from your primary school days, may *not* form part of your final MARV's construction.

- Previous years' MARVs do not count as recyclable material, i.e. you need to construct an entirely new MARV.
- The exception to the recycled materials rule is that the chassis of the MARV may be designed and printed on a 3D printer. Students must provide proof that the design was done by themselves, i.e. the design and manufacturing may under no circumstances be subcontracted. The university has a number of 3D printing resources of which the details will be given on the module's ClickUP site.

- **There is a budget restriction of R500 (in total, i.e. R125 per student) on the components that are used to develop the MARV excluding components and motors that were supplied as part of the hardware tool kit for the module.** 3D printing may be excluded from the budget since 3D printing facilities are available on campus free of charge or at very reasonable tariffs. Components may also be recycled, but to exclude components from the budget, students need to provide proof of their origin, e.g. from an old motherboard, etc. All components that are used must be listed on the budget, but those that are recycled or excluded from the budget may be listed at R0.00.

ii. Assessment and time budget

- **Completion of the preceding practical is a requirement for demonstrating the next one, i.e. Practical 1 needs to be completed and demonstrated to the minimum requirements as specified in the Practical 1 guide before Practical 2 may be demonstrated. In the same way Practical 2 needs to be completed to the minimum requirements as specified in the Practical 2 guide before Practical 3 may be demonstrated. Finally, Practical 3 needs to be demonstrated to the minimum requirements as specified in the Practical 3 guide before the group may participate in the final demonstration, i.e. The Race.**

- The full complement of the practical assignment is designed for a total input of around 280 equivalent lecture periods, which gives around 70 equivalent lecture periods per team member. This includes the mini-poster report and the demonstration. The balance of the module is designed to support the practical project. Do not be tricked into not spending the available time on your MARV in-between practical demonstrations because you do not have to spend much time on other preparation or assignments as in other modules. *The structure of this module differs from that in most other modules in the programs and it is easy to fall into the trap of trying to prepare for the practical assignments in the same manner as for those of other modules.*

- Each student in a team has to participate in both the hardware and software aspects of each practical. Ensure that all team members are familiar with all aspects of the design and implementation of the system. The experience you gain in the practicals will go a long way towards preparing for tests and exams. The more thoroughly you complete the practicals, the less you will need to study for tests and exams. *The teaching strategy in EMK310 is to study as little as possible for assessments in the traditional way, but to gain the knowledge and experience you need through designing, building and testing your MARV with support from the lectures and tutorials.* This is possible since the theoretical and practical components of the module are constructively aligned to offer a unique, effective learning experience.

iii. The team

- *When choosing your team, try to ensure that there is at least one team member with a computer and broad-band internet access in the team to facilitate development of aspects of your MARV that may be completed off-campus.* Much of your MARV may, in fact, be developed off-campus if you include the testing and debugging functionality as recommended in the general methodology section below. Note also that Microchip offers a free cloud-based IDE known as

MPLAB Xpress (http://www.microchip.com/mplab/mplab-xpress) which may be accessed from internet cafes and tablet computers.

- While the full practical assignment is designed for teams of four members, the minimum requirements to pass each practical is designed to be attainable in around 70 equivalent lecture periods, which is the total number of hours that an average student should spend on the practical component of the module.

- Teams may decide to break up and reconstitute if they do not get along simply because we want the experience to be a fun, memorable one. *However, consider that you will most probably have to work with people that you don't like or do not get along with or who do not bring their part in future without the option to avoid the situation.* Rather make use of the opportunity to hone your people skills and try to find an amicable solution to the problem. If you experience friction in your team which you are unable to resolve, please ask the lecturer for assistance to manage the situation.

    - Team changes due to friction among members will not be allowed after Practical 2 as it becomes difficult to take ownership of a new MARV at such a late stage.

    - Teams who have less than four members are encouraged to take in additional team members who may find themselves without a group.

- If you decide to discontinue the module, *please consider your team* and do not discontinue right before a practical demonstration and especially not right before Race Day. Part of becoming an engineer is to develop a responsibility towards the people around you and dropping your team at a tough time is not nice.

c. General methodology to be followed for all practical assignments

The following general methodology is proposed:

a. Study the datasheet of the PIC18F45K20/22 and draw the circuit diagram to realise the system as described above. EMK310 is also known as Datasheets 101.
b. Design and build a hardware circuit to realise the system.
    - *Include a debugging strategy* in the design – you have to demonstrate the debugging functionality that was designed into the system.
    - *Add testing functionality* such as visible LED indicators to allow you to debug your circuit as far as possible without the need for laboratory equipment. Designing this functionality into your circuit also means you understand exactly how it works.
    - Include easy or automated calibration functionality into your system, e.g. by using an LED connected to a level-sensing circuit to indicate when a desired value is reached during the calibration process.
c. Debug the circuit – you cannot test your firmware if you do not know that the hardware is functional.
d. Draw a flowchart to illustrate the firmware execution of the system.
e. Write and compile code in Assembler to implement the system on the PIC18F45K20.
    - *Add testing functionality* that may drive the LED test indicators to allow you to debug your system as far as possible without the need for laboratory equipment.
f. Program the microcontroller, test and debug your system.

d. Demonstrations

i. Logistics

Kindly adhere to the following arrangements to allow scheduled practical sessions to run effectively.

- Demonstrations of the three practical assignments will take place as scheduled.
- **All team members have to be present at the demonstration.**
- All students in a team must be able to answer questions on the complete system.
- No late demonstrations will be allowed.
- *Students must prepare a point-wise demonstration of every minimum requirement for a specific practical as specified in the practical guide.* The requirements will be assessed one by one and only if the team complies with **all** the minimum requirements will they be allowed to proceed with the next practical.
- A scaling factor will be applied to the marks earned for a practical depending on the number of team members.
  - 4-member team: mark is awarded as given in assessment schedule.
  - 3-member team: mark is multiplied by 1.1 *if* the minimum requirements were demonstrated, i.e. the scaling factor only applies to a mark of 6 or higher, up to a maximum of full marks.
  - 2-member team: mark is multiplied by 1.25 *if* the minimum requirements were demonstrated, i.e. the scaling factor only applies to a mark of 6 or higher, up to a maximum of full marks.
  - 1 member "team": mark is multiplied by 1.5 *if* the minimum requirements were demonstrated, i.e. the scaling factor only applies to a mark of 6 or higher, up to a maximum of full marks.
  - However, to have your MARV perform spectacularly in The Race, the minimum pass requirements for the practicals will not suffice. Aim high. We have had single member entries into the Race before who participated in the final and even a single member "team" who won the Race and set a new track record – a marvellous achievement and proof that you are able to do much more than you may realise.
- *Entrance into the next practical is conditional on the successful completion of the previous practical as specified by the minimum requirements for each practical.*

ii. Redemonstrations

Redemonstrations may be allowed if a team has *attended and demonstrated their progress* at the formally scheduled practical session.

- **The constraint to be eligible for a redemonstration is that two of the minimum requirements for a specific practical must be met, of which only one can be a working version of the previous practical (if relevant). The rule of thumb is thus that you have to show some new progress to be eligible for a redemo.**
- Redemonstration of Practical 1 will take place at the latest on the Monday before the demonstration of Practical 2. Redemonstration of Practical 2 will take place at the latest on the Monday before the demonstration of Practical 3. Redemonstration of Practical 3 will take place at the latest before the final demonstration on Race day.
- Redemonstrations will take place during the scheduled practical tutor sessions.

- Only students who comply with the minimum requirements of Practical 3 by its redemonstration date will be allowed to enter their MARV to participate in Race day and will thus be allowed to write the examination for EMK310.
- Successful redemonstration of the system will receive a mark of 50% for the specific practical, irrespective of the mark that would have been awarded had the system been demonstrated at the original practical session.

e. Laboratory notebooks

The protocol for using the laboratory notebook is aligned with the protocol required for Project EPR400/402 that is taken in the fourth year of study.

Since team members have to collaborate but may not always be together when working on the project, one of the following strategies for keeping the labbook is suggested.

a. Since the worldwide governments issued laws stating that all electronic records have the same validity as paper records in 2000, cloud-based electronic labbook that every team member will be able to access and contribute to, are allowed. A shared OneNote notebook works well – every registered student has access to Office 365. You may present an electronic labbook at the practical demonstrations, i.e. you do not need to print it out.
b. If team members do not have access to cloud-based notebook resources all the labbook notes need to be compiled in a single hardcover book into which all the notes have to be glued in an orderly manner.

The following are important:

a. **No hardware or firmware implementation shall be demonstrated without a thorough design, from first principles, being documented in the labbook.** For each missing item that the evaluator requests to see during the practical demonstration, 1 mark from the total practical mark will be deducted. If the final mark is below 5/10 for the practical as a result of the labbook being insufficient, a redemonstration will be awarded and the labbook will be upgraded by the team and must be presented during the appropriate redemonstration session.
b. For hardcopy laboratory notebooks
    - every page in the labbook must be numbered.
    - every entry must start with the date on which the work was done and must be signed off, at the end of the day, by the team members who documented that work.
c. Use the book to document all your thoughts related to the practical, i.e. do not take out pages describing failed attempts. These serve as reference for why things did not work or why you decided to change the approach. The book must document every aspect of the *chronological* development of the project in detail. *Think in your labbook.*

f. Reporting

The reporting strategy for the practical changed in 2015 to (i) introduce students to the concept of keeping a comprehensive and running report of their work, (ii) introduce students to the concept of visual, summarised, essential presentation of their work, and (iii) develop peer-assessment and self-assessment skills in students.

The approach to reports in EMK310 is to primarily assess running documentation that students have created or compiled in their labbooks as part of the development of their MARV and a small poster that summarises the team's work in a way that is assessable by peers. Only the mini-posters are subjected to peer-assessment by means of a *gallery walk*[3] during the practical demonstration time. This strategy saves much time as the documentation should in any case be prepared if students adhere to sound engineering laboratory practices, and facilitates proper evaluation and feedback on reporting. The labbook and mini-poster are prepared in teams.

At the first practical the following reporting components need to be shown.

a. The planning for the project which will be described as "The Plan" (also known as *Project Proposal*) below. This is done in the labbook and is basically a summary of how you plan to approach the project, the specifications that you will be designing for and a concept design. This needs to be done as a team **before physical work on the MARV starts** to allow the team a structured, focused approach to the assignment. You will save yourself much time if you plan thoroughly and the opposite is true too: you will spend many unnecessary hours retracing your steps if you do not plan.
b. Practical 1 final implementation record. This is simply a summary section of documentation in your labbook that shows the final implementation of practical 1. It has to document the specifications that you designed for, the concept of the solution, the detail design and implementation of the hardware and firmware as well as comments, e.g. important notes for integration with the other subsystems of the practical assignment. You may create the summary document with links (cloud-based) or page numbers (labbook). The progress in terms of the time-budget must also be shown. Note: this is not your typical lab report, it is simply a summary of what you finally implemented mainly for your own reference. It may by all means be handwritten with hand drawings where required. If you already documented something well earlier, just add a heading and a note to refer to the page number in the labbook where the details may be found.
c. Mini-poster. Each team needs to prepare an A3 sized mini-poster that gives (i) a one to two sentence summary of the objective of the practical (what you tried to achieve), (ii) the specifications that were designed for (in other words, what values you should measure to prove that the system does what it is supposed to do), (iii) a functional block diagram[4] showing the implementation of the hardware (how the main subsystems integrate with one another), and (vi) a flow diagram showing the implementation of the firmware. Detail hardware designs, i.e. circuits, are not required, especially if the team has a novel approach to protect. The poster must also (v) report the results, i.e. the results must report on the measurements you made to determine whether the specifications were met (don't fudge the results – report it as it is since you are not trying to create an impressive poster, but to make the car run; not achieving the specifications perfectly simply means there is still some work to do), and (vi) present a brief discussion of the results within the context of the entire project.

[3] Each team is given a display spot for their mini-poster on the glass walls and doors in the laboratory after which assessment takes place while walking through the poster gallery.
[4] A functional block diagram is a high-level diagram that shows the functional units of a system. These are the subsystems that can be developed and tested separately, but within the context of the entire system. The sensor system of your MARV may be one of these functional units. In this case the sensor system can be broken down into further subsystems, e.g. the pre-amplification system and the physical sensors. These sub-subsystems may be shown as blocks inside the main functional unit block. The functional block diagram also shows the flow of information and/or power inside the system by the connecting lines between blocks. A functional block diagram contains no detail circuit designs and thus also no components or component values. Each functional block has its own inputs and outputs and is characterised by its own *quantified* specifications.

At the second practical the following reporting components need to be shown.

a. Practical 2 final implementation report. Refer to details for Practical 1.
b. Mini-poster. Refer to details for Practical 1.

Note: no planning report is required as you do this before you start with your MARV. However, if the planning changes significantly, it might be wise to scribble it down somewhere in your labbook..

At the third practical the following reporting components need to be shown.

a. Practical 3 final implementation report. Refer to details for Practical 1. This summary should also include anything that you changed in the first and second practicals to leave a final set of documentation from which your MARV may be replicated.
b. Summary mini-poster. This mini-poster must summarise the entire project and will be displayed at Race Day.
c. Final budget.

i. The Plan (aka *Project Proposal*)

This is a once-off planning report that needs to be presented in the labbook and will be assessed during the demonstration of Practical 1. *The Plan* is for you to help you follow a structured approach. While it will be assessed to force students to follow a proper engineering laboratory protocol, assessment is not the objective. *The Plan* will be at the core of your journey through the development of your MARV and if you neglect this part of the project, you will waste much time and energy on aimless, unfocused work that will hamper your progress rather than support it. *The Plan* needs to consist of the following sections.

Mission critical specifications. What are your designing for? What are the most important specifications that will make your MARV race on Race Day? In formal language: provide and describe the critical specifications that will allow the final system to be functional. An example of one of these specifications could be the accuracy (in measurable terms) with which the sensor system needs to detect the green line. Speed of the MARV may be another mission critical specification since there is a limit on the time allowed for the race. An example of a specification that is not critical is the type of sensor that is used, but the speed at which the sensor can detect the line or the colour of the line may be more important. Also give the source for your choice of specifications, whether it is your own calculations or a reference from literature.

Describe how you will test/prove that each of the mission critical specifications are met, i.e. how will you know whether your MARV is doing what it is supposed to do as well as you designed it for? If there is a sceptic tutor at your demo, how will you prove that your system is working? Remember, no-one can contest a measurement, but an observation that may be interpreted is contestable. For example, if you specify that the MARV must find a specific coloured line in 3 seconds, you can measure and prove the specification. If you say it should find the line fast, your fast and the evaluator's fast may not be the same thing. Another example is to say the MARV must find the line with 99% accuracy versus it must find the line with high accuracy. For the quantified specification you can count how many times it finds the line and present the stats, but the definition of "high" is contestable.

This is an extremely important section in your report since it will affect the final design of your system. If you do not specify your system thoroughly and correctly right from the start, your design will fail. You will have to do some background reading to properly set these specifications.

Hardware concept design. This design is a complete design of your entire hardware system but on a *high* level. You need to break down the system into functional units, e.g. the sensor system, the microcontroller, the motor controller, etc. Each one of these *functional units* forms a system that can be specified and tested on its own.

- Draw a block diagram to show how the functional units connect to and interact with one another.
- List the specifications (measurable) for each of these functional units. These specifications are on a lower level than the mission critical specifications, e.g. the number of sensors you will need to detect the line might be a specification here while it is not a mission critical specification. Indicate the source for the specification, i.e. calculations or literature. Think about your specs: are they realistic within your design?
- Describe for each functional unit the way in which you will test whether it meets each and every one of its specifications.
- Also refer to footnote 5.

Firmware concept design. This design is a complete, though high-level design of your entire firmware (code). You need to break down the firmware into concept states (refer to state machine tutorial), e.g. data acquisition (getting data from the sensors), processing and control. Each one of these states should be designed to be specified and tested on its own. This state machine will form the basis for your firmware design and will help you to design with integration of the various parts of the system in mind.

- Draw a concept state machine to show how the states connect to and interact with one another.
- List the specifications (measurable) for each of these states. These specifications are on a lower level than the mission critical specifications, e.g. the sampling rate of the data acquisition state. Indicate the source for the specification, i.e. calculations or literature. Think about your specs: are they realistic within your design?
- Describe for each state the way in which you will test whether it meets each and every one of its specifications.

Gantt diagram. Plan the time and tasks that you will perform on a day-to-day basis for the entire semester. *Remember, you do not get much preparation or homework in EMK310, i.e. you need to plan this time into the practical preparation.* You need to show your progress on this diagram at each practical demonstration/redemonstration, i.e. we will refer back to this original planning at each practical demonstration session. Allow space where you can plot actual progress below planned progress on this diagram.

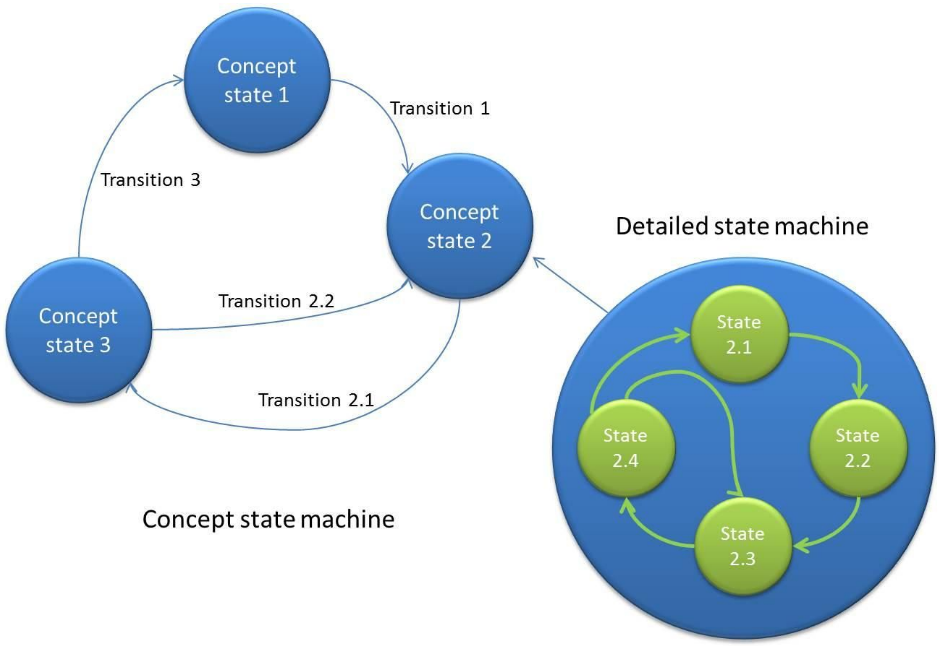
ii. Final implementation labbook record for each practical

The detailed, chronological documentation of all aspects of your system will form the basis for this report. **Compile a summary record for each practical.** This record shall be the last entry in your labbook at the time of the practical demonstration, but may contain references to pages in your labbook where a specific aspect of the system, e.g. the hardware detail design, is reported. The summary record must contain the following.

Hardware detail design. Provide the final detailed design for each functional unit that is demonstrated at the specific practical. Give both block diagrams and detailed circuit diagrams for the functional units.

Hardware results. Tabulate or graph the detail specifications of each functional unit and the measured results that prove that the specifications have been met. You may be asked to demonstrate the verification of any specification, i.e. be ready to repeat measurements during the practical demonstration. If a specification has not been met, give reasons and propose a course of action to correct the system.

Firmware detail design. Provide a detailed state machine for each the concept states that you showed in your firmware concept design. From this compile a detailed flow diagram and/or pseudocode to describe the functionality of each detailed state machine. The basic idea behind this approach is provided in the figure below.



Firmware results**.** Tabulate the detail specifications of each state and the measured results that prove that the specifications have been met. You may be asked to demonstrate the verification of any specification, i.e. be ready to step through your code and repeat outcomes during the practical demonstration. If a specification has not been met, give reasons and propose a course of action to correct the system.

Time planning. Compare your initial time planning with the actual time expenditure on the project. Do this on your original Gantt diagram.

Budget. Compile a list of components and their prices to prove adherence to the budget limit.

iii. Mini-poster presentation for each practical

Each team needs to prepare an A3 size mini-poster for each of the three practical demonstrations, presenting the concepts behind the different subsections of that specific practical as well as the results obtained. The poster template will be available as a separate file to download from ClickUp.

Formatting requirements:

- Font size not less than 14 point so that the poster is readable at a distance of 2 metres.
- The use of pictures and photo's with supporting text to explain concepts is recommended, i.e. an all-text poster will not be appealing. Font size on pictures should not be less than 12 point.
- *No handwritten posters*, i.e. posters need to be prepared on a computer and printed out. You may laminate your poster if you prefer, but this is optional.
- Two copies are required. The first is to be displayed at the workbench in the lab where the team demonstrates their practical. The second is to be displayed on a wall in the laboratory on a designated position for peer-assessment.
- **A softcopy of the poster needs to be uploaded to ClickUP by 14:00 on the day of the practical demonstration.**

Assessment:

- **All hardcopy posters need to be up at the start of the practical session, i.e. at 14:30 on the day of the practical demonstration.**
- The mini-posters will be peer-assessed and evaluator-assessed making use of a "gallery walk" approach.
- Each practical team will receive a list of three randomly selected teams whose posters they need to assess. The different assessment criteria will be explained in class and will be made on a four point scale, from 0 to 3 where 0 indicates a poor attempt, 1 indicates an adequate attempt, 2 indicates a good attempt and 3 indicates an excellent attempt.
    - Take care to assess your peers' posters as you would like your poster to be assessed.
    - Be honest: the skill to assess a piece of work is important in our environment. This exercise develops your critical thinking and evaluation skills. Try to be objective. You are not doing someone a favour by giving them high marks they do not deserve, and neither will you be able to keep down a good team by awarding a low mark for good work.
    - Your assessment will be weighed against the evaluator-awarded marks (see point below) to determine whether you are fair in your approach. Teams will not be penalised if they award unrealistic marks, but their mark may be disqualified from the assessment at discretion of the lecturer.
- The evaluators will use the same assessment criteria as those for peer-assessment to award a mark for a team's poster.
- The final "report" mark for a specific practical will consist of the average of the peer-assessment marks, which will contribute 50% of the mark and the evaluator-awarded mark, which will contribute the other 50% of the mark.
- You need to record your marks for each team in an online form that will either be administered through a Google form or through a ClickUP assignment.

Redemonstrations:

- Students need to upgrade both their labbook reports and their mini-posters for a redemonstration, i.e. you need to submit new documentation for a redemonstration showing your progress since the last demonstration. The same rules for documentation as those for demonstrations apply, e.g. penalization if documentation is incomplete, etc.

- The report mark for a redemonstration will be 50% if all reporting components comply with the pass requirements.

Practical 1: Talking and testing

MARV's communications, data transfer and storage, and test protocol

1. Introduction

The development of your MARV is divided into three stages and thus into three practical assignments of which this is the first. In this practical you will develop (i) a communication system that will implement serial communication in the form of RS232 and I²C protocols, (ii) transfer and store data between the PIC18F45K22 and a serial EEPROM, and (iii) design and implement an integrated unit test/LED-based test protocol to allow you to debug, test and demonstrate the communication, data transfer and storage systems without an oscilloscope.

2. Learning objectives

The objectives of this practical are to familiarise the student with

- the software development tools that accompany the PIC microcontrollers,
- writing program code in assembly language,
- basic program flow,
- programming of a microcontroller,
- the implementation and debugging of interrupts and delays,
- the use of I/O functions and ports,
- the use of timers,
- the use of seven segment displays (SSDs), and
- the implementation of a simple application with a PIC microcontroller.
- I²C, serial and USB interfaces,
- external data storage,
- communication between a microcontroller and a PC,
- communication between an external EEPROM and a microcontroller,

3. Assignment

a. Communication, data transfer and storage subsystem (CDTS subsystem)

Implement a USB to serial bridge to facilitate communication between a PC and the EUSART of the PIC18F45K22 using the development board that you received. Communication must also be established between the PIC18F45K22 and an <u>external</u> serial EEPROM using I²C .

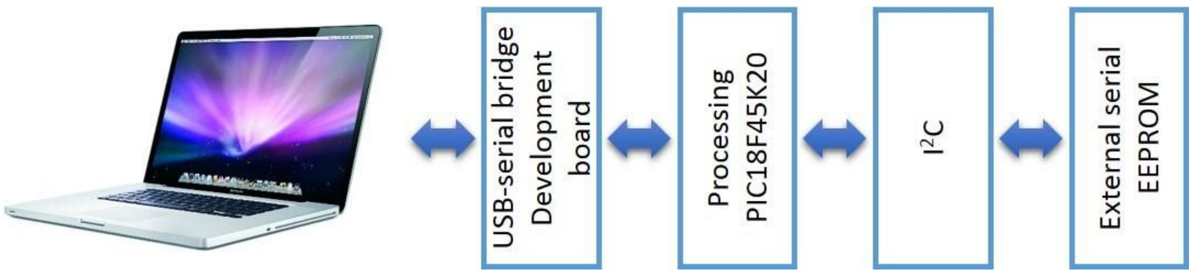


Figure P2.1. System to realise serial communication between PC and external memory device.

Figure 1 shows a block diagram of the system that needs to be designed. A serial terminal program (PUTTY, or similar available on all laboratory computers) will be used to communicate between the serial port of the PC and the PIC18F45K22 via a PIC18F14K50 USB-to-serial bridge. The PIC18F45K22 will also communicate with an external EEPROM.

The communication channels will provide a user-interface to program the MARV into different modes of operation.

The user will use the following commands to control data flow among the various components using the serial terminal program:

Command/Mode	**Action**
MSG<Enter>	This command enables programming of MARV's start-up message. The message (at least 5 characters), must be stored in the external EEPROM and must be returned from the microcontroller to the serial interface upon start-up of the system. In other words, if the serial port is connected to the PC and you switch on the power, the start-up message must be read from the EEPROM and displayed in the terminal application. When the MSG command is issued, the EEPROM must be cleared and reprogrammed with the new message that follows the enter. Notes: ● Upon normal start-up, the message must be returned and the system must enter RACE mode. ● The SSD must display "0" while MARV is in MSG mode. ● This command is only valid if the system is in RCE mode.
PRC<Enter>	This command enables **p**rogramming of the **R**ace **C**olour and strategy. The microcontroller must return "What shall MARV race? " when the command is given. Once in PRC mode, the user may proceed to program the desired Race Colour or Race Strategy: B for racing on blue track R for racing on red track G for racing on green track n for racing on black track L for Maze Racing[5] Notes: ● The SSD must display "1" while MARV is in **PRC** mode. ● This command is only valid if the system is in **RCE** mode.
RCE<Enter>	This command places the system into RACE mode. The microcontroller must return "MARV races X" (where X is B, R, G, n or L) when the command is given. This should be the default state of the system. Notes: ● The **RCE** command is issued to return the system to RACE mode if the system is in one of the other two modes (**MSG** or **PRC**). ● You may choose any default Race Colour/Strategy (B,R,G,n,L) for MARV to start up in. ● The SSD must display "2" while MARV is in RACE mode, but not racing. ● The SSD must display the Race Colour/Strategy (R,B,G,n,L) while MARV is racing.

[5] More about Maze Racing later.

CAL<Enter>	This command places the system into CALIBRATION mode. When in CAL mode, the system must cycle through the different colours and record the sensor reading for each colour. White ( ) and black ( ) must be included. Notes: ● At this stage, CAL mode will be implemented as a CALIBRATION framework, which will be expanded to be fully functional for the second practical. ● The system must start cycling through each colour. The SSD must show the colour for which calibration must be performed (B,R,G, or), after which either an automatic or manual calibration must be performed. ● As soon as the system is calibrated for a colour, it needs to light an LED to indicate successful calibration. This LED may be one of the LEDs on the "dump port" described in the test protocol below. ● For this practical the physical calibration sequence must be replaced by a delay of 3 seconds, i.e. the system must cycle through a sequence of display-colour-on-SSD, delay (in place of the real calibration protocol), indicate successful calibration with LED, for all five colours.

Communication protocols

Communication between PC and PIC
Type: RS323 asynchronous serial (3-wire cable)
Rate: 19 200 Baud proposed, but you may choose
Format: ASCII
Bits: 8-bit; one start bit; one stop bit; no parity

Ensure that you specify the correct baud rate in your serial terminal software. Set "flow control" to "none" and ensure that the correct COM-port is selected (you can check this in the PC's Device Manager that may be accessed through the Control Panel's "System" tab).

Communication between PIC and EEPROM
Type: I^{2}C (refer to datasheet).
NB: The I^{2}C protocol need not be hardcoded, i.e. you may make use of the on-board I^{2}C functionality of the PIC.

b. Hardware test protocol and firmware unit tests

Design and construct an integrated firmware unit test[6]/LED-based hardware test protocol that will indicate

- whether serial communication to the PIC was successful,
- whether serial communication from the PIC was successful,
- whether I^{2}C transmission from the PIC was successful,
- whether I^{2}C reception by the PIC was successful.

Develop a subsystem that uses an external interrupt to dump the content of a register to one of the ports.

[6] Unit testing is a software development process in which the smallest testable parts of an application, called units, are individually and independently scrutinized for proper operation. Unit testing can be done manually but is often automated. [Source: searchsoftwarequality.techtarget.com/definition/unit-testing]

- The register that needs to be dumped must be coded into the interrupt service routine for the test, i.e. it can be changed depending on the subsystem that is being debugged.
- The output of the port must be visualised using LEDs to facilitate debugging.
- The LEDs on the port must double as colour indicators, i.e. they must be marked to indicate track colours for debugging purposes. Different colour LEDs may also be used instead of text markers.

Also refer to the broad assignment for additional specifications.

4. Minimum requirements

At least two of the minimum requirements must be met to earn an opportunity to redemonstrate Practical 1. **All of the minimum requirements must be met to pass the practical and to be allowed to proceed to the next practical.**

- Two-way serial communication between the PC and PIC18F45k22 via a USB-to-serial bridge must be demonstrated by having the PIC8F45K22 echo the commands as specified for the CDTS subsystem back when it receives them through the USB serial link to the PC.
- The system must be able to ***write and read*** a string consisting of at least two characters (e.g. "AB") consistently to and from the EEPROM.
- The testing system for the RS232 and I^2C serial communication channels must be operational and integrated with the two serial communication systems.
- The SSD must be able to display the characters as required.
- The calibration system framework must be operational according to specification (cycle through colours with correct delay).

5. At the demonstration

- The system must start up in **RACE** mode.
- First the RS232 serial communication implementation must be demonstrated including the testing protocol followed by the same for I^2C.
- Next the functionality of all the different commands and modes must be demonstrated.
- Finally, the functionality of the calibration system must be demonstrated.

Students will need to answer questions about the hardware and firmware implementation of this assignment.

6. Components required

- PIC18F45K22 microcontroller (provided)
- One digit seven-segment display unit, e.g. SA36 (to be purchased by the student)
- Development board included in class fees
- I^2C EEPROM (provided by Microchip)
- Various electronic components, e.g. resistors, capacitors and LEDs (to be purchased by the student)
- Oscilloscope cables to facilitate displaying of signals at the demonstration

7. Grading

Grading will be as follows:

Mark [10]	**Requirements**

Fail	Fewer than two of the minimum requirements as stated in paragraph 4 above can be demonstrated.
Redemo	Two to four of the five minimum requirements as stated in paragraph 4 above can be demonstrated.
6	**Minimum requirements** as stated in paragraph 4.
7 to 8	Functionality of RS232 or I^2C above minimum requirement can be demonstrated. 1 mark for each.
9	Almost perfect
10	Perfect

Practical 2: Sensing and starting
MARV's colour sensor, navigation algorithm and touch start

1. Introduction

In this practical you will realise the colour sensor (CS) subsystem that is required to detect green, blue, red and black lines on a white background and stop the MARV at the end of the track. The optimal approach is to integrate the sensor developed for this purpose in ENE310 into the MARV. You will also develop a navigation algorithm (NA) that is based on your MARV's sensor system as well as the touch start (TS) sensor that will start your car at The Race. Furthermore, you will integrate the CDTS subsystem and the testing protocols of Practical 1 with the subsystems developed for this practical. Calibration of the colour sensors is included in the assignment and must be integrated into the calibration framework of Practical 1.

NOTE: Students who are not enrolled for ENE310 will need to develop the colour sensor array/matrix as part of this practical assignment while students who are enrolled for ENE310 will have the opportunity to develop the sensor array/matrix in the first practical assignment of ENE310. While students are allowed to redesign the colour sensor for EMK310, this is not a desirable approach since it will amount to repeating something that was intended/designed as a once-off effort.

The integrated sensor system is the single most important subsystem in your MARV that will determine whether your car will race or rove.

If the sensor system is inadequate, the entire project will fail. There is an important engineering rule that says *Junk in, junk out*. This means that you cannot post-process junk data to give you good data to base your control strategies on. Junk data will result in junk control. However, if you can read good, trustworthy data from your sensors, implementation of the control strategy to navigate your MARV will be much easier than when you have to contend with erratic data.

This practical together with the first ENE310 practical form the foundation of the project from an implementation point of view and students must take great care in thoroughly designing the sensor system (as opposed to constructing a system by trial and error). You must also design the sensor system with its final purpose in mind, i.e. do not waste time to implement a sensor system that will not be able to provide the data that you need to navigate your MARV.

2. Learning objectives

The objectives of this practical are to familiarise the student with

- analogue-to-digital conversion and signal conditioning,
- calibration strategies,
- touch sensors,
- system integration.

3. Assignment

a. Colour sensor system

Design, construct, calibrate and test a colour detection system that can distinguish between green, blue, red and black on a white background to follow a coloured track as described in Figure 1. The system must make use of an interrupt (any one) and an interrupt service routine. The sensor design will be covered in ENE310. However, your MARV will not be able to navigate with only one sensor, i.e. you need to make your design modular so that you may create a sensor array or matrix to allow (i) detection of crossings in time for the car to start turning when necessary and (ii) following of the track, i.e. to allow centring of the car over the track. The specific configuration of sensors will be one of your team's design challenges to give your MARV an edge over your opponents' MARVs. Remember, once again, that the design of your sensor system at this stage will determine the success of your MARV in the final race. For this reason teams are advised to *meticulously* design the sensor system of their ARVs.

The design and implementation of any electronic circuitry to condition the different sensors' output signals and which are not developed as part of the ENE310 practical are included in the assignment. The design of the conditioning circuits for the sensors must be given in your lab book. The design specifications must be given and their values have to be justified[7].

b. Navigation system

The navigation system needs to implement a control algorithm based on the output from the sensor array, i.e. it needs to demonstrate at least the concept states of the control algorithm's state machine. This should consist, at a minimum, of the following: (i) line detected, drive straight, (ii) line bends to left, turn left, (iii) line bends to right, turn right, (iv) cross-over detected, either change course or stay on track (depending on what you plan to do) (v) line lost, execute search algorithm, (vi) end of track detected, stop. Each state must be represented by an LED on your circuit that is labelled with the condition that it represents. You may use the LEDs that are part of the debugging system for this function. The labels must make it easy for the evaluator to assess the functionality of the navigation algorithm. In addition there must be green, red and blue LEDs on your car that show the colour of the line that it is following at that stage. When the car loses the track, i.e. it is not positioned over a red, blue or green line, the three LEDs must switch on and off one after another (sweep) continuously with an on time of 333 ms per LED (1 s to sweep through all three LEDs).

c. Calibration system

A calibration system to tune the sensors to the track's colour is of major importance. Minor colour variations in track colour, different lighting conditions, reflections, etc. may affect the operation of the sensors thus demanding a calibration system. The system should allow you to calibrate the

[7] Justified in this context means you need to be able to give a reason, based on scientific calculations and engineering reasoning of why you choose a specific specification. For example, if your car drives at a specific speed, the sensors must be able to provide a stable reading within a specific time which is related to the speed and which you will be able to calculate.

sensors on site within one minute. You should aim to minimise, through careful planning and design, the effect of factors that may upset the sensor system so that minimal calibration is required.

d. Touch start subsystem (TS subsystem)

Design, construct and test a subsystem that uses the ADC on the PIC18F45K20 to sense when a person touches a metal disk. The TS subsystem will be used to start your ARV during the final race. There is a Microchip application note available that describes this type of touch switch.

e. Integration of subsystems

Integrate all the subsystems of Practicals 1 and 2 such that the system starts a race when the metal disk of the TS subsystem is touched by a person, i.e. touching the sensor is the "go" signal for your MARV. Since the MARV does not have motors yet, you will manually move it along the track for this practical.

Also refer to the broad assignment for additional specifications.

4. Minimum requirements

At least two of the minimum requirements must be met to earn an opportunity to redemonstrate the practical. All of the minimum requirements must be met to pass the practical.

- All of the minimum requirements for Practical 1.
- Correct detection of green, red and blue versus black and white and display thereof through the three coloured LEDs.
- Demonstration of the interrupt that will be used as well as execution of the ISR associated with it. Consistent detection of this interrupt and execution of interrupt service routine must be demonstrated for at least 5 successive interrupt conditions. Any of the relevant interrupt sources, e.g. an ADC interrupt if the ADC is used, may be demonstrated as long as it is the interrupt that will be used in the sensor system.
- Demonstration of a 333 ms delay (10% error acceptable) on an oscilloscope, i.e. accurate visualisation of the delay. Either a delay loop or a timer implementation is acceptable. *This needs to be displayed on the oscilloscope and not in simulation since i) it is difficult to determine the delay by inspection and ii) the correct implementation in hardware needs to be verified.*
- Demonstration of rudimentary navigation functionality, specifically states (i),(ii), (iii) *and* (v) as described in the navigation assignment above. This includes the visual indication of the states through different LEDs that indicate the current state of the system (drive, turn left, turn right, search). This requirement will necessitate integration of several of the subsystems. This requirement will also require a sensor array consisting of at least three elements.
- Correct detection of the end of the track (state (vi) in the Navigation system above). The end of the track must be detected to stop the ARV automatically to prevent you MARV to run away on Race Day.
- The TS system needs to be functional.
- Integration of the CS, navigation and TS subsystems.

At the demonstration

Each of the minimum requirements described above must be demonstrated by manually moving your MARV over a test track (since it will not yet have motors). The test track will be of the same material with the similar properties (except for trajectory and length) than those of the competition tracks. The MARV must be constructed so that it does not damage the tracks, e.g. by scraping of the colour with sharp edges on the sensor system.

Students will need to answer questions about the hardware and firmware implementation of this assignment.

5. Components required

- Subsystems of Practical 1 as specified in minimum requirements for Practical 1
- Optical sensors, e.g. OPB706 (reflective sensor); red, blue green LEDs with photo-transistor or photo-resistor (to be purchased by the student)
- Small conducting metal disk (to be purchased by the student)
- Various electronic components, e.g. resistors, capacitors and LEDs (to be purchased by the student)
- Oscilloscope cables to facilitate displaying of signals at the demonstration

6. Grading

Grading will be as follows:

Mark [10]	Requirements
Redemo	Minimum requirements as stated in paragraph 4 not attained.
6	**Minimum requirements** as stated in paragraph 4.
7	Minimum requirements plus integration of one more function than required for the minimum requirements.
8	Two details not quite perfect
9	One detail not quite perfect
10	Perfect

Practical 3: Start your engines!
Motor control, system integration, battery charger and fine-tuning for performance

1. Introduction

In this practical you will develop the subsystems that are required to control the motors that drive and navigate your MARV along the track. You will integrate the subsystems developed for Practicals 1 and 2 with the subsystems developed for this practical.

2. Learning objectives

The objectives of this practical are to familiarise the student with

- Motors, motor drivers and controllers
- Basic control systems
- System integration

3. Assignment

a. Motor control and navigation

Design, construct and test a system that uses a PIC18F45K20/PIC18F46K20/PIC18F45K22 to control the motor drivers (which may be constructed from first principles or purchased off-the-shelf) that are required to drive the motors required for translation (i.e. to make the ARV drive in a straight line) and rotation (i.e. to make the ARV turn). If drivers are purchased off-the-shelf, an analysis of the circuits that are implemented in these drivers must be included in the report. The motors that were provided must be used, but may be adapted (e.g. gear ratio) to improve performance. The code required to drive the motors must be developed from first principles.

b. Vehicle construction

Design and assemble a simple chassis and install motors with wheels to drive this structure/vehicle. Everything besides the wheels, electronics and motors has to be constructed from recycled materials. The chassis of the vehicle may be designed from first principles and printed on the 3D printer facilities on campus. Proof that the design was done by the team members will need to be submitted. The vehicle may not be larger than 100 mm x 180 mm x 120 mm (W x L x H) and must use one or two rechargeable 9V batteries as a power source.

Two possible translation and rotation methods are:

i. one motor is used for translation (driving forward) and the other for rotation (turning) of the ARV, or
ii. two motors are used to separately drive the two front wheels and the difference in rotation speed between two motors is used to control translation and rotation.

c. Battery charger

A battery charger needs to be implemented to charge the 9V batteries. The basic approach is to connect the batteries to a power source, to measure the charge current and to disconnect the source from the batteries as soon as the charge current drops below a predefined threshold, i.e. the battery voltage approaches that of the charging source. An additional PIC16 microcontroller is provided for the charging circuit. The PIC18F45K22 may alternatively also be used to implement this function.

Also refer to the broad assignment for additional specifications.

4. Minimum requirements

At least two of the minimum requirements must be met to earn an opportunity to redemonstrate the practical. All of the minimum requirements must be met to pass the practical.

1. All of the minimum requirements for Practicals 1 and 2.
2. The translational motor must be functional, i.e. you must demonstrate that the motor can be controlled through the microcontroller.
3. Basic integration with the TS subsystem, i.e. the motor(s) must be controlled to rotate the wheels in one direction when the go signal is given from the TS system.

4. Basic integration with the CS subsystem, i.e. the ARV must stop automatically when the end of the track is detected by the CS subsystem.

Requirement in addition to minimum requirements (does not count towards redemonstration minimum requirements):

5. Budget in labbook with proof of expenses.

Minimum recommended outcome

1. The requirements for practical 3.
2. A vehicle chassis and wheels with a mounted translational motor that drives the wheels.
3. Control electronics mounted on the MARV chassis that enables the ARV to drive in a straight line (the first section of the track is straight, so a team can at least earn 1 mark for the race even if their MARV cannot navigate).

5. At the demonstration

Refer to the "At the demonstration" sections of Practicals 1 and 2 since many of these requirements will be required to test the final functionality of the system. For this practical the ARV will start in RACE mode and be placed on a short section of track. The ARV's driving and navigation performance will be assessed.

Students will need to answer questions about the hardware and firmware implementation of this assignment.

6. Components required

- Subsystems of Practicals 1 and 2
- Maximum two motors (any) that will enable the vehicle to drive and navigate (included in component kit provided to students).
- Various electronic components, e.g. resistors, capacitors and LEDs (to be purchased by the student).
- Components for a motor driver (to be purchased by the student).
- Power source, e.g. a battery that must fit onto the ARV and power all subsystems and motors (to be purchased by the student).
- Recycled materials for the chassis of the ARV (to be salvaged by the student) or 3D printed chassis according to specifications.
- Oscilloscope cables to facilitate displaying of signals at the demonstration

7. Grading

Grading will be as follows:

Mark [10]	**Requirements**
Redemo	Minimum requirements as stated in paragraph 4 not attained.
6	**Minimum requirements** as stated in paragraph 4.
7 to 8	Rudimentary autonomous navigation can be demonstrated.

9	Almost perfect
10	Perfect

Favourite student sayings

Student at practical on Teflon spacers that were required to keep two Vero boards apart: "Mam, these little spacers were so expensive, they are like the Gucci of components."

Student upon finding out that most line-follower robots described on the internet are designed for black and white detection: "Prof, we stopped going to Google. Google is useless!" (Mission accomplished☺.)

Student upon cornering the lecturer in a hallway and asking to assist with Race Day: "Prof, even though I did EMK three times, it was my favourite module ever."

Student upon hearing the plans for the following year's race: "I won't even mind to fail EMK; it sounds so exciting!"

Student from other department walking past the first Race Day preparations: "This is epic!"

Spectator asking about Race Day: "Is this an international competition?"
Prof T: "Indeed, all the students in the world that are enrolled for EMK310 participate in the competition."

2017 Race Day winner Petri Oosthuizen: "Work hard and follow your passion."

Race Day

"Would you like me to give you a formula for success? It's quite simple, really. Double your rate of failure. You are thinking of failure as the enemy of success. But it isn't at all. You can be discouraged by failure or you can learn from it. So go ahead and make mistakes. Make all you can. Because remember that's where you will find success."

— Thomas J. Watson

The Rules

i. All specifications and arrangements in this document are applicable.

ii. The MARV must be started via the TS subsystem and must stop automatically at the end of the track.

iii. The MARV must navigate to collect the greatest number of points (by activating the greatest number of sensors) in the shortest possible time.

iv. The order for placing of winners will be accuracy (highest score which is equivalent to the number of sensors activated) followed by speed (fastest time first).

v. No team member will be allowed on the track.

vi. There is a time limit of three minutes on completion of the race.

vii. An external timing system will be used to time the duration of each vehicle's race.

viii. If a vehicle does not reach the end of the track, the result will be the number of sensors that were activated during the race.

ix. There will be eight identical tracks on which teams will compete in heats.

x. The sixteen MARVs that complete the track within the maximum three minutes with the highest score in the shortest times will be entered into the two semi-final races.

xi. The eight MARVs that complete the track in the semi-finals within the maximum three minutes with the highest scores in the shortest times will be entered into the final.

xii. Prof T reserves the right to enter a ninth MARV in the final race based solely on her discretion.

xiii. One invited Old Champs race will be driven by past final-race MARVs. This race will be eligible for its own prizes.

xiv. Only teams who's ARVs complete the race with a score of more than four will be eligible for letters of recommendation.

xv. Only students of teams who participated in the race will be eligible for entry into the exam in EMK 310.