

Wacky Racers 2022 Instructions

Michael Hayes

Version 1.1, February 23, 2022

1 Introduction

The purpose of this assignment is to design, build, and program an embedded system using an ARM microcontroller and surface mount technology.

The goal for each group of four students is to build a remote controlled vehicle (the Wacky Racer) and its controller (the Wacky Hat). At the conclusion of the assignment there will be a dastardly race!

Each group is comprised of two sub-groups of two students. One of these subgroups constructs the Wacky Racer and the other constructs the Wacky Hat. You may be asking why is the Wacky Hat called the Wacky Hat? Well, a hat that controls a remote vehicle using head motions is not an ordinary hat!

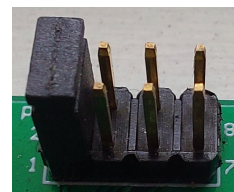
2 Requirements

The following requirements are mandatory if you wish to maximise your marks.

2.1 Wacky racer

1. The chassis is to be constructed by each group. These can be 3-D printed, constructed from Perspex or wood, etc. A standard chassis is available from the Electronics Lab technician (Diego Ramirez). The electronics must be visible on top of the chassis.
2. Have a standard working bump sensor (supplied).
3. Locomotion can only use two 6 V DC motors (supplied).
4. Everything must be powered from a single 5-cell NiMH battery pack (supplied).
5. Use a single four layer printed circuit board of dimension 85 mm×64 mm.
6. Use an ARM microcontroller (Atmel SAM4S8).
7. Drive the motors using H-bridges (Texas Instruments DRV8833 dual H-bridge is recommended).
8. Regulate the nominal battery voltage to 5 V with a buck regulator IC (ADP2302ARDZ-50).
9. Be decorated with an LED tape (supplied) controlled by the MCU.
10. Use a USB interface for debugging.

11. Use a serial wire debug interface for MCU programming/debugging.
12. Have adequate battery fusing and reverse polarity protection.
13. Have a sleep pushbutton to enter low-power mode.
14. If the battery voltage drops below 5 V, an LED should flash and high power draw devices should be disabled.
15. Have an active-high red LED to indicate errors.
16. Have an active-high green LED to indicate status.
17. Interface to the Wacky Hat with a Nordic nRF24 SMD radio module.
18. Have four jumper selectable radio channels.
19. Be humorous.



Each Wacky Racer can have a dastardly means of hindering another team's Wacky Racer. However, you cannot:

- Damage or destroy another Wacky racer (except in the battle royale)
- Damage the venue
- Injure a spectator
- Jam the radio signals

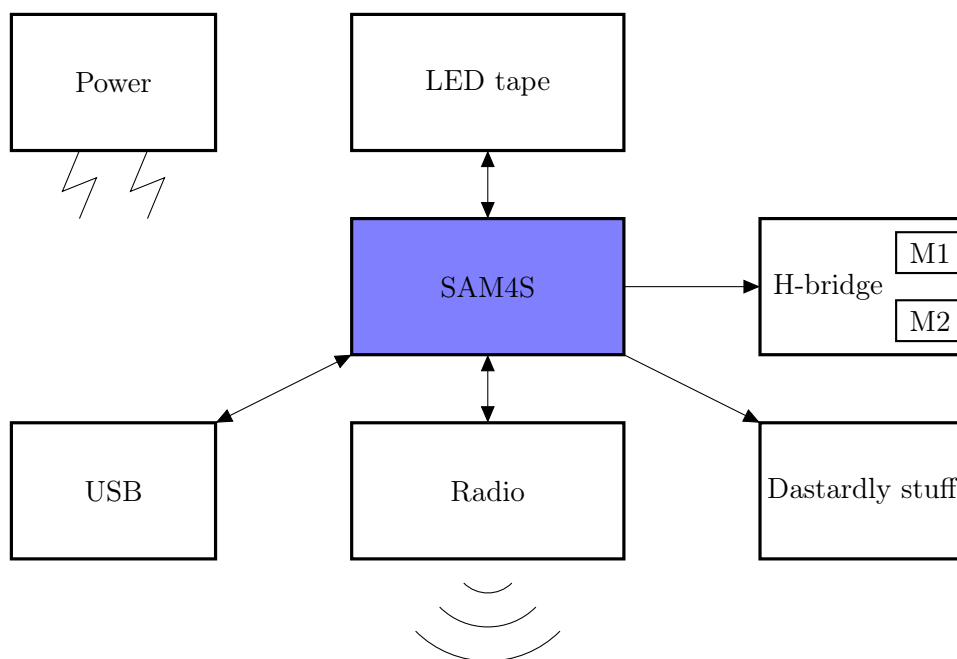


Figure 1: Racer board top level diagram.

2.2 Wacky hat

1. Construct a Wacky Hat that contains all the electronics.
2. Everything must be powered from a single 5-cell NiMH battery pack (supplied).
3. Have adequate battery fusing and reverse polarity protection.
4. Use a single four layer printed circuit board of dimension 85 mm×64 mm.
5. Use an ARM microcontroller (Atmel SAM4S8).
6. Regulate the nominal 6 V battery voltage to 5 V with a buck regulator IC (ADP2302ARDZ-50).
7. Be decorated with an LED tape (supplied) controlled by the MCU.
8. Use an I2C IMU (MPU-9250) for head motion detection.
9. Use a USB interface for debugging.
10. Use a serial wire debug interface for MCU programming/debugging.
11. Have a joystick in case the IMU does not work.
12. Have a sleep pushbutton to enter low-power mode.
13. If the battery voltage drops below 5 V, an LED should flash and high power draw devices should be disabled.
14. Have an active-high red LED to indicate errors.
15. Have an active-high green LED to indicate status.
16. Play sound when the bumper is hit.
17. Interface to the Wacky Racer with a Nordic nRF24 SMD radio module.
18. Have four jumper selectable radio channels.
19. Be humorous.

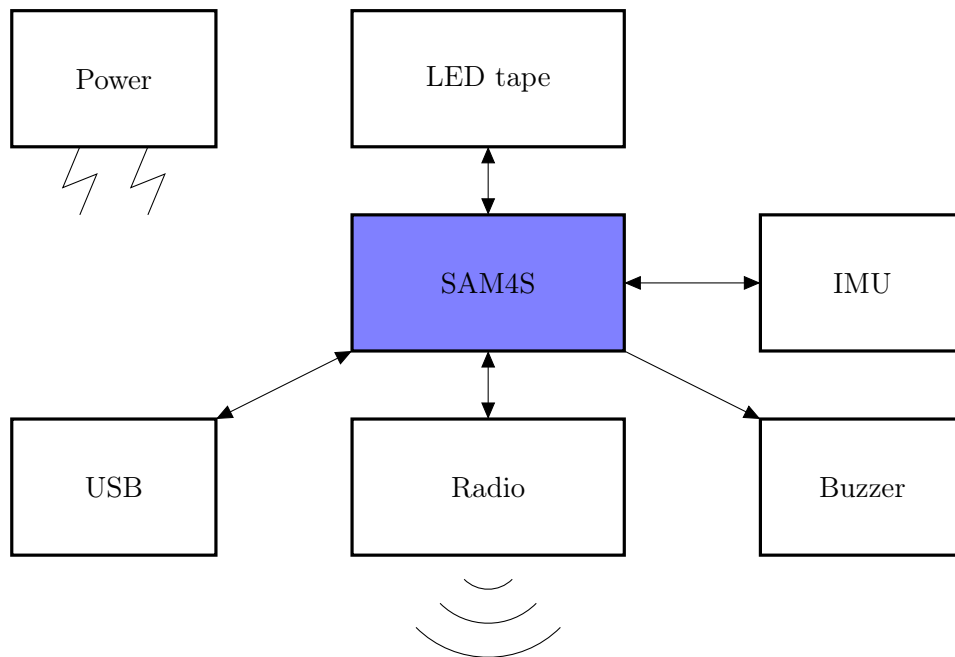


Figure 2: Racer hat top level diagram.

3 Assignment schedule

The planned activities for the timetabled labs in the Embedded Systems Lab (ESL) are:

Week	Assistance	Assessment
T1-1	Altium tutorial 1 (schematics)	Schematic submission for review (Friday 5pm)
T1-2	Altium help	
T1-3	Schematic review	
T1-4	Altium tutorial 2 (PCB)	
T1-5	Altium help	PCB submission 1
T1-6	Altium help	PCB submission 2
T1-7	General	
B-1	(break)	
B-2	(break)	
B-3	(break)	
T2-1	General	Blinky
T2-2	General	IMU/motors
T2-3	General	Radio control
T2-4	General	Functionality
T2-5	Competition	Competition, critique

Notes:

1. There are two lab streams. Your group will need to sign up to one of the streams. Choose a group between 1–19 for the Tuesday stream or group 20–39 for the Wednesday stream.
2. There may be a 6–10 day delay for the PCBs to be manufactured from the time of submission. You will then need to book an assembly session in the SMT lab provided you have done the SMT lab induction.
3. Do not underestimate the blinky milestone. It requires having a functional PCB, a functional toolchain, and the ability to download code into the microcontroller's flash memory.

4 Assessment

The marks breakdown (max. 0x64) is:

PCB submission	0–0xa marks
Blinky milestone	0x5 marks
IMU/motor milestone	0x5 marks
Radio control milestone	0x5 marks
Functional assessment	0–0x14 marks
Board inspection	0–0x1e marks
Competition	0–0xa marks
Individual critique	0–0xf marks

4.1 Milestones

There are five milestones. To achieve the associated marks, they must be demonstrated to a T.A. by the end of the lab for your allotted stream. If you need an exception to this, see Daniel Morris with a *very* good reason. The milestone requirements are:

Schematic review Submit your A3 schematic on Learn for review. Lose 10 marks if you miss the submission time.

PCB submission Submit your PCB design to Learn for manufacture. To encourage early submission there is a sliding scale for marks depending on when the PCB is submitted, see table. After week 6, there is a 10 mark penalty per week. **NB, a rushed PCB design will cause you more grief, more PCB rework, and a lower mark for the inspection.**

Week	Submission day	Cut-off time	Mark
5	Monday	1.00 pm	10
5	Tuesday	1.00 pm	9
5	Wednesday	1.00 pm	8
5	Thursday	1.00 pm	7
5	Friday	1.00 pm	6
6	Monday	1.00 pm	5
6	Tuesday	1.00 pm	4
6	Wednesday	1.00 pm	3
6	Thursday	1.00 pm	2
6	Friday	1.00 pm	1

Blinky Demonstrate that can blink an LED controlled by the SAM4S.

IMU/motors For the Wacky Hat, demonstrate output of IMU readings to a PC using USB CDC. For the Wacky Racer, demonstrate control of the motors from a PC using USB CDC.

Radio control Demonstrate sending commands from the Wacky Hat to the Wacky Racer over a radio link.

If you cannot show the functionality of a previous milestone during any assessment, you will fail that assessment and loose any marks from the previous milestone.

4.2 Functionality assessment

Functionality requirements:

Wacky racer	Wacky hat
Blink LED	Blink LED
Drives LED tape	Drives LED tape
Drive motors forward/backward	Read from IMU
Speed control of motors	Calculate speeds from IMU
Steering control	Joystick control
Receive radio message	Send radio message
Jumper selectable radio channel	Jumper selectable radio channel
Dies on bump	Plays sound on bump
Low voltage indication	Low voltage indication

Marks are allocated on how well things work. Up to 5 bonus marks can be awarded for extra functionality such as:

Wacky racer	Wacky hat
Dastardly stuff	Plays interesting sounds
Sleep mode	Sleep mode

Sleep mode is where you shutdown the MCU to save energy and use a pushbutton to awake the MCU via an interrupt. Is is easy to get wrong and hard to save every microamp.

4.3 Competition

The competition is a race around an obstacle course. Marks will be awarded every time you pass over a Wacky Ramp in the correct order.

To be awarded any marks for the race:

1. Your vehicle must stop for at least 5 s if the bumper is hit.
2. Your vehicle must be controlled by motions of the Wacky hat sitting on someone's head.

After the races, there will be a battle royale where the last operational wacky racer wins. Hitting the bumper of an opponents wacky racer removes them from the battle.

Marks will also be awarded for wackiness, group costumes, etc.

4.4 PCB inspection

This is assessed after the competition. The categories are:

1. Layout (component placement etc.)
2. Construction (tidiness, rework, etc.)
3. Testability
4. Power supplies (routing, decoupling, etc.)

5 Schedule

Week 1

- Form a group of four and register your group on Learn. If you cannot form a group of 4, contact Daniel Morris.
- Read the Requirements section in this document.
- Read the System Design section in the The Wacky Racers Guide.
- Attend Altium schematic tutorial.

Week 2

- Start your schematic design.
- Peruse the datasheets of the key components.

Week 3

- Submit your Altium schematic for review.
- Attend review feedback session.

Week 4

- Attend Altium PCB tutorial.
- Start your PCB layout.

Week 5

- Watch SMT lab induction video.
- Submit PCB design (early round).

Week 6

- Submit PCB design (late round).

Week 9

- Populate and test PCB.

Week 10

- Populate and test PCB.
- Demonstrate blinky program.

Week 11

- Demonstrate blinky program.

Week 12

- Demonstrate IMU/motors functionality.

Week 13

- Demonstrate radio control.
- Plan your costumes!

Week 14

- Demonstrate full functionality.
- Prepare your costumes and dastardly stuff!

Week 15

- Practice control of your wacky racer.
- Attend wacky race.
- Submit PCB for inspection (box in SMT lab).
- Return chassis, batteries, and ST-link programmers.
- Submit individual critique to Learn.

6 Technical stuff

Read this section carefully. There are clues as to how we mark your PCBs.

6.1 Version control

Use version control for everything, or else! Learning git is frustrating but is a skill you will not regret, see The Wacky Racers Guide for details.

Your group leader should create a forked copy of the wacky-racers git project and then add the other group members to the project. This can be done by:

1. Go to <https://eng-git.canterbury.ac.nz/wacky-racers/wacky-racers>
2. Click ‘Fork’ button. This will create a copy of the main repository for the project.
3. Click on the ‘Settings’ menu then click the ‘Expand’ button for ‘Sharing and permissions’. Change ‘Project Visibility’ to ‘Private’.
4. Click on the ‘Members’ menu and add group members as Developers.
5. Using a bash terminal (or other useful shell), enter the command:

```
$ git clone https://eng-git.canterbury.ac.nz/your-userid/wacky-racers.git
```

If you do not want to have to enter your password for every git push/pull operation, you should set up ssh-keys and use the git URL instead:

```
$ git clone git@eng-git.canterbury.ac.nz:your-userid/wacky-racers.git
```

6. Add a remote URL for the main repository.

```
$ cd wacky-racers
$ git remote add upstream https://eng-git.canterbury.ac.nz/wacky-racers/wacky-racers.git
```

Again if you do not want to manually enter your password, you can use:

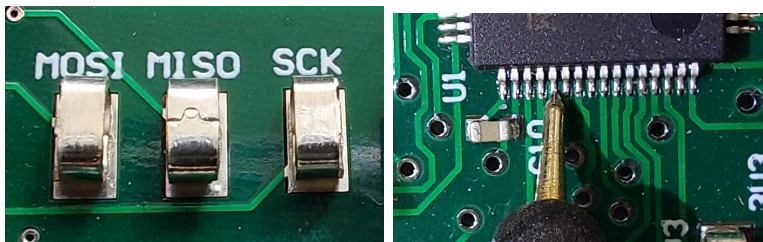

```
$ cd wacky-racers
$ git remote add upstream git@eng-git.canterbury.ac.nz:wacky-racers/wacky-racers.git
```

7. If we add more demo code or tweak the instructions in the main repository, you can get the updated stuff using:

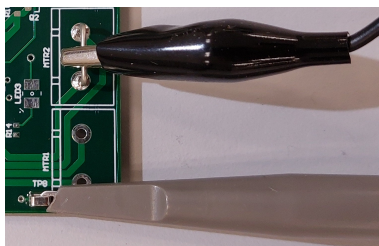
```
$ git pull upstream master
```

6.2 Schematics

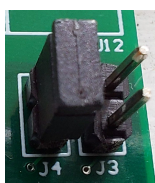
1. Read the Altium tutorial on Learn.
2. Read the The Wacky Racers Guide.
3. Add you and your partner's name and your group number to the title block.
4. Save PDF files of your schematics in your source repository. **Note, when debugging your PCBs, we will not help you until you show us your printed A3 schematic.**
5. I bet that you will not have enough test points to clip an oscilloscope probe to. Do not think you can hold the probe tip against an MCU pin without shorting something. Ensure you give a meaningful name to the test point in its comment field.



6. Ground test points are essential for an oscilloscope earth clip. A U-shaped piece of wire soldered into two holes is good for this. Keep ground test points clear of other test points since the earth clip may short between them. You will probably require at least two ground points on opposite sides of the PCB.



7. Checking the schematic is the most crucial part of the assignment. If the schematic is wrong then your PCB will be wrong and you will need to do rework. So, schematics must be thoroughly checked by another person.
8. It would be useful to have a jumpers connected to PIO pins so that you can configure your board. For example, if a jumper is in, use the joystick, otherwise use the IMU.



9. In case you forget something, I suggest running some spare PIO pins from the MCU to some pads or a connector.

6.3 PCBs

1. Your four-layer PCBs are going to be manufactured in batches. There is at least a week turnaround time to get the boards made.
2. It is important that you check footprints for parts they you create. We will impose a 10% penalty for each rerun of a PCB, say due to a footprint mistake. Get your partner to check.
3. In Altium, turn off the test point designators and turn on comments instead. This lets you give them human readable names to make your lives easier.
4. PCB layouts must be thoroughly checked by another person.
5. A PCB track can blow faster than a fuse. So keep high current tracks fat and short.
6. Clearly mark the positive and negative battery connections on the silk screen.
7. Some of the chips can get hot so thermal considerations are required. Follow the manufacturers' guidelines in the datasheets.
8. The switching regulators can interfere with the radios.
9. Use a design rule check to see if any of the following constraints are violated:
 - Minimum trace width (0.15 mm)
 - Minimum trace clearance (0.15 mm)
 - Minimum via size (0.3 mm hole, 0.6 mm outer diameter)
 - Minimum hole size (0.3 mm)
 - Minimum annular ring (0.1 mm)

For every violation of one these rules, we will deduct 1% from your final mark.

10. Check the checklist in the The Wacky Racers Guide before submission.

6.4 Assembly

1. Finding shorts is extremely frustrating so maximise clearances and test for shorts before populating components.
2. Components can be put through the oven on the reverse side although heavy components may need to be glued.
3. Never assume where pin 1 is on an IC; check the datasheet. 5–10% of groups will get this wrong.

6.5 Software

1. Read the The Wacky Racers Guide.
2. If you are not using version control, you are foolish.

3. Inspect MPH's sample code.
4. If you are trying to program the SAM4S for the first time and are feeling tired or impatient, then do something else.

6.6 Debugging

1. Start running small programs (such as the provided demo programs) to test each feature separately.
2. An oscilloscope is your friend.
3. It is possible to use the GDB debugger but you need to know what you are doing, especially with optimised code.
4. Drawing a diagram of what you think is happening is highly recommended. A simple circuit diagram or timing diagram will often help you realise what you have missed and let you fix it without asking for help.

6.7 Possibly asked questions with answers

- *Why use the SAM4s MCU?* For this application most MCUs would suffice, even an 8-bit AVR microcontroller. To level the playing field, I have chosen a MCU most students would not have used before. This is an ARM based MCU made by Atmel I have used this in a number of projects. Indeed we used to teach this chip in ENCE361. There are many other similar MCUs made by different manufacturers such as the STM32 that would just as suitable.
- *Why use a four layer PCB?* Come to lectures to find out!
- *Why use 7.2 V NiMH batteries for the Wacky Racers?* These were a legacy of previous Wacky Racers. They are also safer than lithium batteries with sleep-deprived students.

7 Assistance

Daniel Morris (<mailto:daniel.morris@pg.canterbury.ac.nz>) is in charge of the assignment with support from the technicians Scott Lloyd (<mailto:scott.lloyd@canterbury.ac.nz>) and Diego Ramirez (<mailto:diego.ramirez@canterbury.ac.nz>), and a team of teaching assistants.

- Emails to the lecturers (except of a personal nature) will be quietly ignored.
- If you have a generic question, please use the ENCE461 Learn discussion page. Note, under the advanced options you can send your post without a 30 minute delay.
- All decisions regarding legality of your dastardly devices are at the whim of Daniel Morris. Email Daniel Morris if you wish to keep your ideas secret.
- TAs will be available in the scheduled lab times. Priority will be given to groups assigned to the current lab session. **We will only provide assistance to students who have a printed A3 schematic sheet in front of them and have already tried looking up the problem in the The Wacky Racers Guide.**
- For questions pertaining to Altium, surface mount assembly, and surface mount rework, see Scott Lloyd (SMT Lab technician) and Diego Ramirez (Electronics Lab technician).

- Michael Hayes is really busy pulling all the string behind the scenes and so will only help with gnarly problems referred to by Daniel Morris!

8 COVID

- Some of the lab sessions may need to be held online.
- The first term does not need lab access, however, you will need to be able to run Altium remotely or run the free student-edition of Altium. The instructions and files can be found on the ENCE461 Learn page.
- If because of illness or isolation your group cannot meet a milestone, email Daniel Morris.
- Some of the milestones may need to be altered.