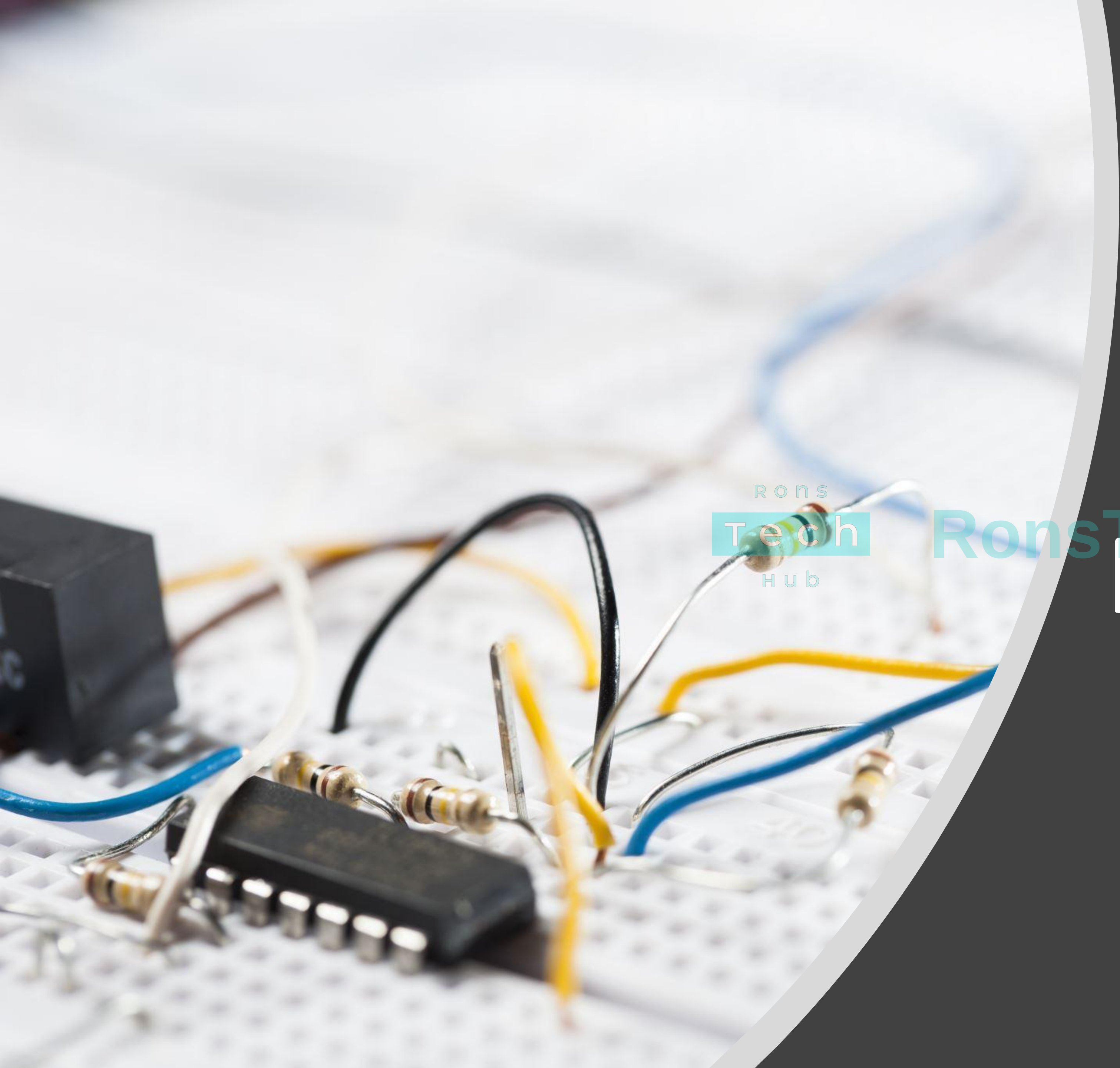


Unit 6 Microcontroller

Introduction

Course Information

- The name of the Unit is:
- Unit 6 – Microcontroller Systems For Engineers.
- We are now engineers and need to understand how these systems work.

Course Information

- This is an exam unit. A practical exam.
- It is a 12 hour exam.
- There is no coursework.
- This is a big unit.
- 120 Guided Learning Hours.

Specification

This has everything you will ever need to know for Unit 6.

You should have a look at this.

Specification Link.

How Will You Be Assessed?

- There are four objectives of this unit.

- You will need to meet all of these to get a Distinction.
- You will be given a task to complete in the 12 hours.
- This will be spread across multiple days.

Typical Grade Boundaries

- These can change from year to.

- They tend to not change massively.

31725H: Unit 6 Microcontroller Systems for Engineers

Grade	Unclassified	Level 3			
		N	P	M	D
Boundary Mark	**0**	7	15	34	53

Past Paper 2022

Past Paper 2022 Internal Link

Please ensure you read the paper carefully.

Feel free to make highlights as there is a lot of information.

Submission

Three things are needed for submission.

A PDF, Electronic Task Booklet.

A video file of the no longer than **three minutes**.

A fully completed authentication sheet.

Electronic Booklet Submission

Naming convention if my name was Ron Builder.

booklet_[Registration number #]_[surname]_[first letter of first name]

Booklet_200111_builder_r

This MUST be a PDF.

Video File Submission

Naming convention if my name was Ron Builder.

file_[Registration number #]_[surname]_[first letter of first name]

file_200111_ builder_r

Authentication Sheet Submission

This should be given to you by the examiner or the college.

Please ensure all the details are filled in correctly.

Template Document

- Link To Template Document.
- Use this as a guide.
- It will have examples of how to complete each activity at the distinction level.
- If you want to work from this document, download a copy.

Exam Help Booklet

This was created by the previous lecturer here.

It combines both his examples and the examples from the examiner.

Link To The Document.

All Activities – PDF Submission

- A single PDF **MUST** to be given at the end.
- You use a Word Processor or PowerPoint.
- I recommend a Word Processor.
- You will possibly loose marks if a PDF is not given.

Learning Aims Of Course

1

Produce a technical specification and a design for a microcontroller system to solve a problem.

2

Develop and test a software and hardware solution for a microcontroller system to solve a problem.

3

Project manage the development lifecycle and present the operation of a microcontroller system to solve a problem.

Exam Activities

There are a total of six activities.

Each activity leads into the next.

Exam Activities

Activity 1 - Task Planning and System Design.

Activity 2 - Analysis Of The Client Brief.

Activity 3 - System Design.

Activity 4 - System Assembly and Programming.

Activity 5 - System Testing and Result Analysis.

Activity 6 - System In Operation.

Rons
Tech
Hub

RonsTechHub

Exam Activity 1

- Task Planning and System Design.
- Simply plan what you are expected to do.
- This section will be update along with all the others.
- It will hold information on what was done in each section and why.
- Time: 1.5 Hours.

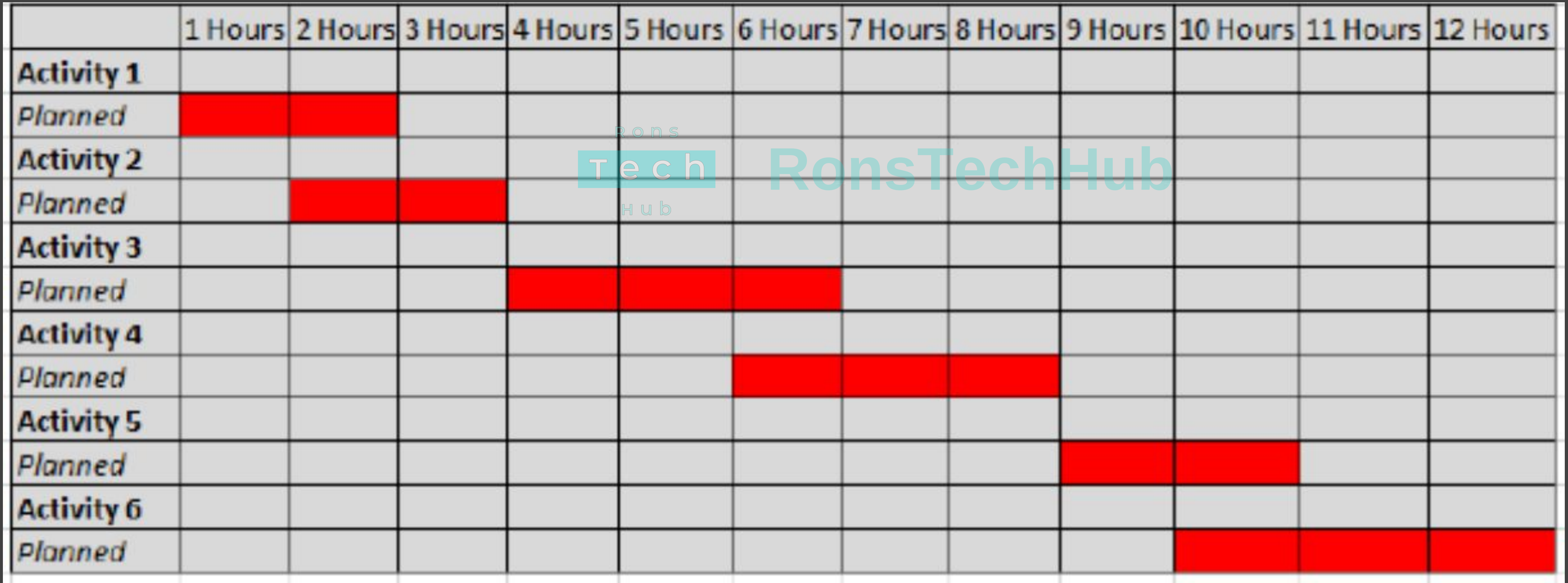
Activity 1 – Plan

- You will need a plan of the work.
- Do a timeline on what you **INTEND** to do.
- The plan can also be named a Gantt Chart.
- You will have two versions. Before and after.
- For this we can use Microsoft Excel.

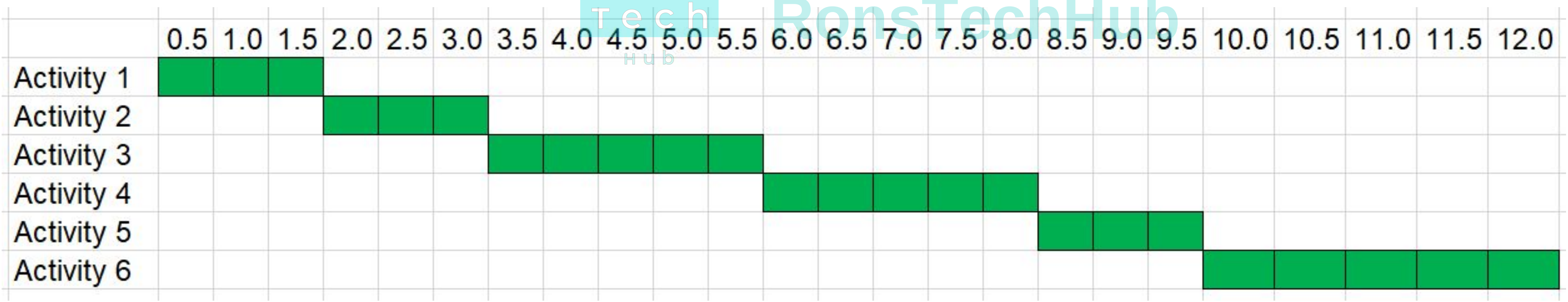
The Plan

- This is simply what you INTEND to do.
- Things will most likely change.
- You will have the chance to come back and make an ACTUAL log of what you did.
- Some of you might do two logs, some might do a single overlapping log.

Gant Chart Example 1



Gantt Chart Example 2



Expected Times

- You get the recommended times from the exam paper.
- Just put those times in.

- If it does change later, that NEEDS to be reflected.
- It most likely will change.

Two Gantt Charts

One will be before you start working.

It should be a plan of what you intend to do.

The other will be the **actual** time taken to each section.

Daily Log For Exam

After the Initial Gantt Chart the next to do is to create a log.

The log should ideally have the following points.

All the logs MUST be in Activity 1.

Points For Daily Logs

- Date.
- Day of exam.
- What have I done this session?
- Issues encountered this session and solutions with justification.
- Action points for next session.

Daily Logs...... Again

- You **MUST** always come back to the logs.
- Complete all the previous points into each log for each day you have the exam.
- Do one log per day.
- If your exam lasts for four days, you will **NEED** 4 logs.

Blank Daily Log

- IF your exam lasts four days, you will NEED four of these.
- If the exam lasts for five days you will NEED five.
- <u>This is in the share document.</u>

Daily Log

1

Think back to the three Pi Pico mini projects you did.

2

Pretend they were all on the same day.

3

Complete a daily log for those.

Daily Log

Google Classroom Lesson

Daily Log Document.

Examiners report showing:

- Gantt Chart
- Daily Log

Recap and Activity 1 Practice

- Use the blank document provided to you. Link To Document.
- Fill it in using the below:
- Make the Pi Pico LED blink.

- Attach variable resistor to Pi Pico and get readings in Thonny console.
- Get a temperature sensor reading.

Recap and Activity 1 Practice

- Make a plan using excel (Gantt Chart).
- Leave enough space for the updated Gantt Chart.
- Make a copy of the daily log.

- Fill in the daily log using the information from the previous slide:
  - Make the Pi Pico LED blink.
  - Attach variable resistor to Pi Pico and get readings in Thonny console.
  - Get a temperature sensor reading.

Exam Activities Timing

- Activity 1 - 1.5 Hours.
- Activity 2 - 1.5 Hours.
- Activity 3 - 2.5 Hours.
- Activity 4 - 2.5 Hours.
- Activity 5 - 1.5 Hours.
- Activity 6 - 2.5 Hours.

Video for Gantt Chart

Video for Day 1 Log

Rons Tech Hub RonsTechHub

Exam Activity 2

- Analysis Of The Client Brief.
- Read and understand what should be done overall.
- Create requirements.
- Time: 1.5 Hours.

Activity 2

- You need to read the client brief and FULLY grasp what needs to be done.
- You essentially need to detail all the things the client wants you to do.
- Have a look at the Activity 2 section from the Examiner's Report.
- Use that as a starting point.

Activity 2

- A combination of the client's needs and your knowledge to enhance the system in which ever ways you deem fit.

- This is essentially an interpretation of the client brief.

- Again, your own understanding plus possible additions or improvements to make the system better.

Hardware For Exam

- PC with USB port.
- Arduino or **Raspberry Pi Pico**.
- I have opted to use the Raspberry Pi Pico (W).

Software For Exam

- Windows or Linux PC.
- Word Processor.
- Spreadsheet Processor.
- C Programming Language/Arduino IDE/Thonny with MicroPython.
- Fritzing circuit Design and or Microsoft Visio.

From Template

- By interpreting the client brief into operational requirements, prepare a technical specification for a user-friendly system that can handle some unexpected events.
- Prepare a test plan to check the functionality of the final solution against the technical specification and include some unexpected events.

Client Brief

- Not client's underwear. Haha joke.
- This is what the client wants you the engineer to do.

- You can think of this as a Design Brief or Design Document.

Requirements

- There are different types of requirements.
- You need to focus on the Operational Requirements.
- What needs to be done.
- Why it needs to be done.
- What will be used to do it.
- Google Search: Operational Requirements Definition

Operational Requirements

- As an engineer, think of ways to produce the system.
- Think of ways to make the system better. Possible enhancements.
- Eg: Why would it be better to have an LCD showing a count rather than a single LED flashing when a count increases?
- The above links to the user-friendly nature of the system.

Operational Requirements

Unexpected events are just that, things that are not supposed to happen in an ideal world.

We must make the system able to handle it these.

Typically, this can be linked to health a safety issue or a way to stop/restart the system if there is an error.

Having a STOP button/switch that will halt the entire process until someone can check the system could be useful.

Technical Specification

Prepare a technical specification for a user-friendly system that can handle some unexpected events.

A technical specification, specifies/describes all technical procedures related to product development.

Google Search: Technical Specification.

Technical Specification

- [Google Search: How to Write A Technical Specification?]
- Explains what a product or project will do and how you will achieve these goals.

Read Client Brief

Technical Specificaiton

This can be a simple list.

Or.

It can be a series of paragraphs.

Once the detail necessary is present, the choice is yours.

Technical Specification – 2022 Paper

- One90 Mobile Accessories the client has designed a magnetic car mount. This magnetic car mount as its name states has a mount with a magnet attached to it.

- The client brief describes a system to be designed which checks if a magnet is present in the body of a mount device. The magnet will sit atop the mount body.

- Simply put the system checks if the magnet is there or not. If the magnet is present the device should continue along the production line. If the magnet is not present the device should eb removed from the assembly line.

Technical Specification – 2022 Paper

- The system needs to be able to count the products which are good and which are faulty. When the good products get to a count of 10 there should be a notification. It would make sense to do the same for a count of 10 for faulty products.

- A single LCD, two LEDs and a single buzzer can be used for the notification system. The LCD will clearly display the count in real time. LCD example:
  - Good Pro = 11
  - Bad Pro = 5

Technical Specification – 2022 Paper

- A **red LED** could be used to flash x times when a count of 10 **faulty** products has been reached.

- A **green LED** could be used to flash x times when a count of 10 **good** products has been reached.

- A single buzzer could be used to give a **positive** beep sound to flash alongside the green LED.

- The same buzzer could be used to give a **negative** beep sound to flash alongside the green LED.

Technical Specification – 2022 Paper

- A simple text file could be used to save the count of the system this would be saved to the system's storage to ensure that the information is still available even if the power is removed. Each count session will have the date and time saved along with the good and faulty counts. Appending the information means the old information will not be lost.

- An emergency pressure pad/push to close button could be used to stop the system in the event of an emergency. This would save count and indicate emergency button was pressed in the text file.

- Another button could be used to restart the system taking it back to the main function.

Abbreviations and Keywords

- LCD – Liquid Crystal Display.
- LED – Light Emitting Diode.
- Hall Sensor – Detects a magnetic field.

- Magnometer – More sensitive compared toa hall sensor.
- Buzzer – Sounds device with limited range of sounds.
- Push to Close – A button which closes the loop of a circuit or wiring.

Rons
Tech
Hub

RonsTechHub

Testing

A test plan is just that.

What do you INTEND to test.

How do you intend to test it.

The result you are **expecting**.

Testing

Before you can test the system, you need to have a rough idea on how the system will be designed.

You MUST have already gone through the requirements of the system.

What is it required/need to do?

You test against the requirements and sub requirements.

Testing

You need a system to do the following:

Turn on the lights and open the blinds when someone enters a room.

Close the door behind the person when they get to the desk.

Testing Lights and Blinds

- Purpose of the test:

- To check whether the blinds open and the lights turn on when a person has been detected in the room.

Testing Lights and Blinds

Test Condition:

A motion detection device should trigger the motor for the blinds to be opened and turn the lights on.

The system should either poll periodically or always be on.

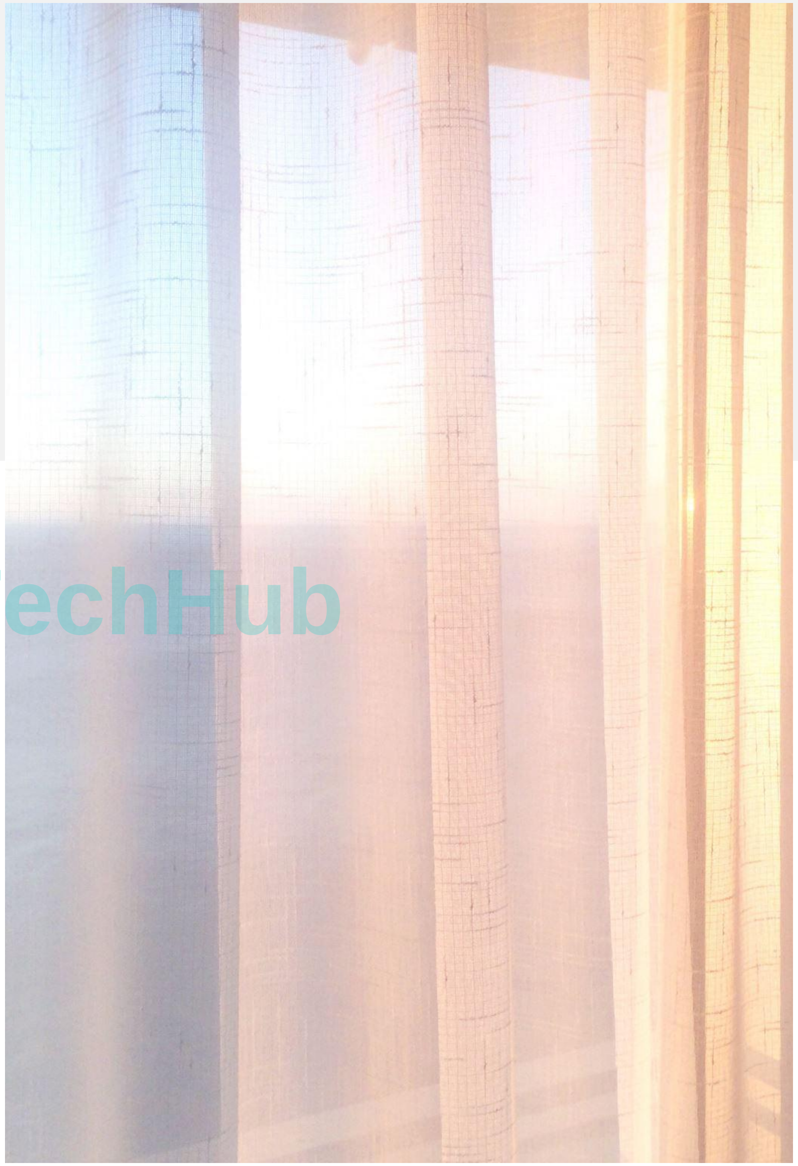
If motion == 1 for example the motor for the blinds should start moving in the desired direction AND the Lights should go on.

Testing Lights and Blinds

- Expected Result:

- The system should lay dormant. Nothing should happen until a sufficiently large body enters the room.

- Once a presence has been detected the system should carry out desired actions of turning on the lights and opening the blinds.



Testing

- All your tests MUST be related to the system you are asked to design.
- Learn/understand how to do a test plan for all systems.
- Do NOT memorise test plan for a single type of system.
- Test plans can be applied to any system given to you.

Update Activity 1

- I THINK activity 1 and some of activity 2 can be done in a single day.
- You need to go back to your activity 1 log and update all that was done.
- Ensure that the tasks done and the logbook all tie in.

Video on Test Plan

Rons
Tech
Hub

RonsTechHub

Exam Activity 3

- System Design.
- Diagrams, drawings and circuit diagrams showing how the system should look and work.
- Pseudocode.
- Time: 2.5 Hours.

Activity 3 From Template

- Prepare a user-friendly system design that can handle some unexpected events, including:
  - The selection and **justification** of suitable input and output devices.
  - A description of the system design covering input and output devices and microcontroller connections.
  - A plan for the program structure detailing key system operations.

Design

- Easily one of my favourite sections, I like to draw diagrams and work out system logic.
- You can do several things for this section.
- Google Search: Pesudocode
- Google Search: Flowcharts

System Design

- You need to start considering the inputs and outputs for the system.
- Exactly what they might be.
- Why are they important, you MUST justify why that component.

System Design – From Examiner

- Learners should also describe the function of the input and output devices.
- This should also include the microcontroller connections.
- That means the pin that the component is connected to.
- In this section you can be as technical as possible.

System Design Components

- List all the components and give the following information:
- Part Name.
- Purpose of the Part (why is it needed).
- Input or Output Device.
- Quantity Needed (and explain why).
- This can be a table or a paragraph for each.

Components For Motion Sensed Lights

- Part Name: PIR (Passive Infrared Sensor).

- Purpose of Part: Passive Infrared Sensor detects motion by detecting radiation levels that are emitted from living creatures. This is to be the main trigger for the system.

- Input/Output: This is an input device.

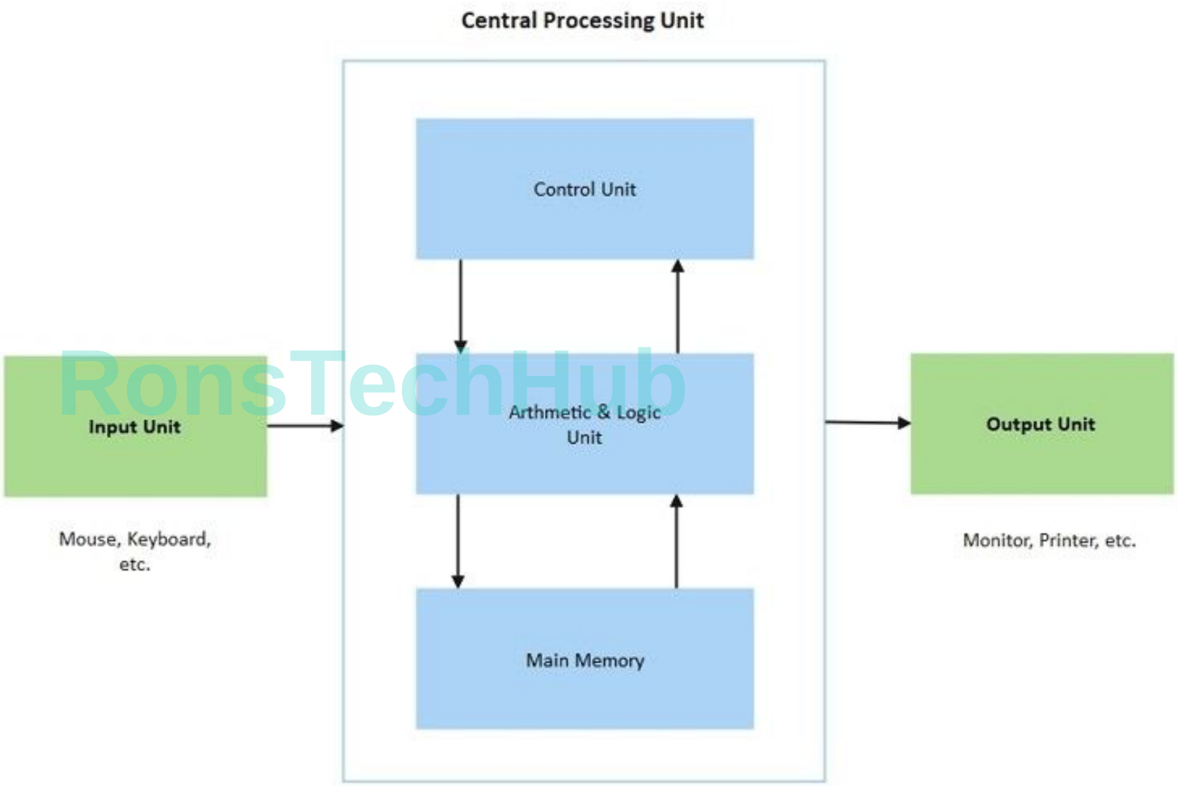
- Quantity: Only one is needed to trigger the system.

System Diagrams

- A block diagram, showing only the inputs, processor and output.
- A system diagram which can be a circuit diagram or a fritzing circuit diagram.
- The circuit diagram will show all the detail of how things should be connected.

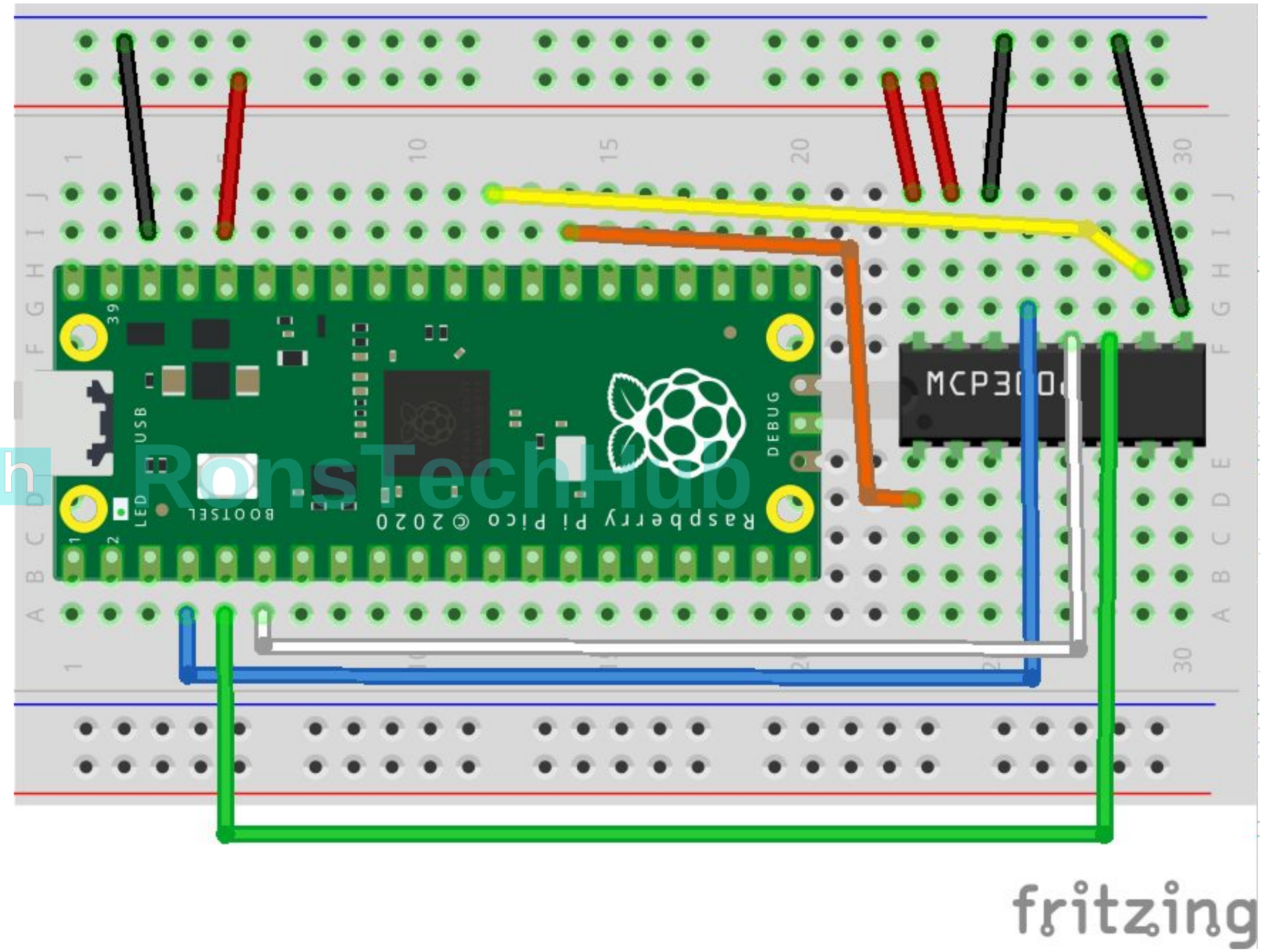
Block Diagram Example

- A simple block diagram which shows:
- Input.
- Processing.
- Output.



Circuit Diagram Example

- A fritzing diagram is fine.
- If you use another program that is also fine.
- If you can do it from scratch in a computer design program like photoshop that is fine.
- I highly recommend Fritzing.
- It has Arduinos, Raspberry Pis, PICs and many more.



Pseudocode and/or Flowchart

- You need to produce pseudocode or a flowchart.
- The choice is yours.
- Make the design as detailed as possible.
- Flow charts take more time to create, keep that in mind.

Pseudocode Key Words

- BBC Bitesize: Pseudocode

- **INPUT** – indicates a user will be inputting something
- **OUTPUT** – indicates that an output will appear on the screen
- **WHILE** – a **loop** (**iteration** that has a **condition** at the beginning)
- **FOR** – a counting loop (iteration)
- **REPEAT – UNTIL** – a loop (iteration) that has a condition at the end
- **IF – THEN – ELSE** – a decision (**selection**) in which a choice is made

Pseudocode Example

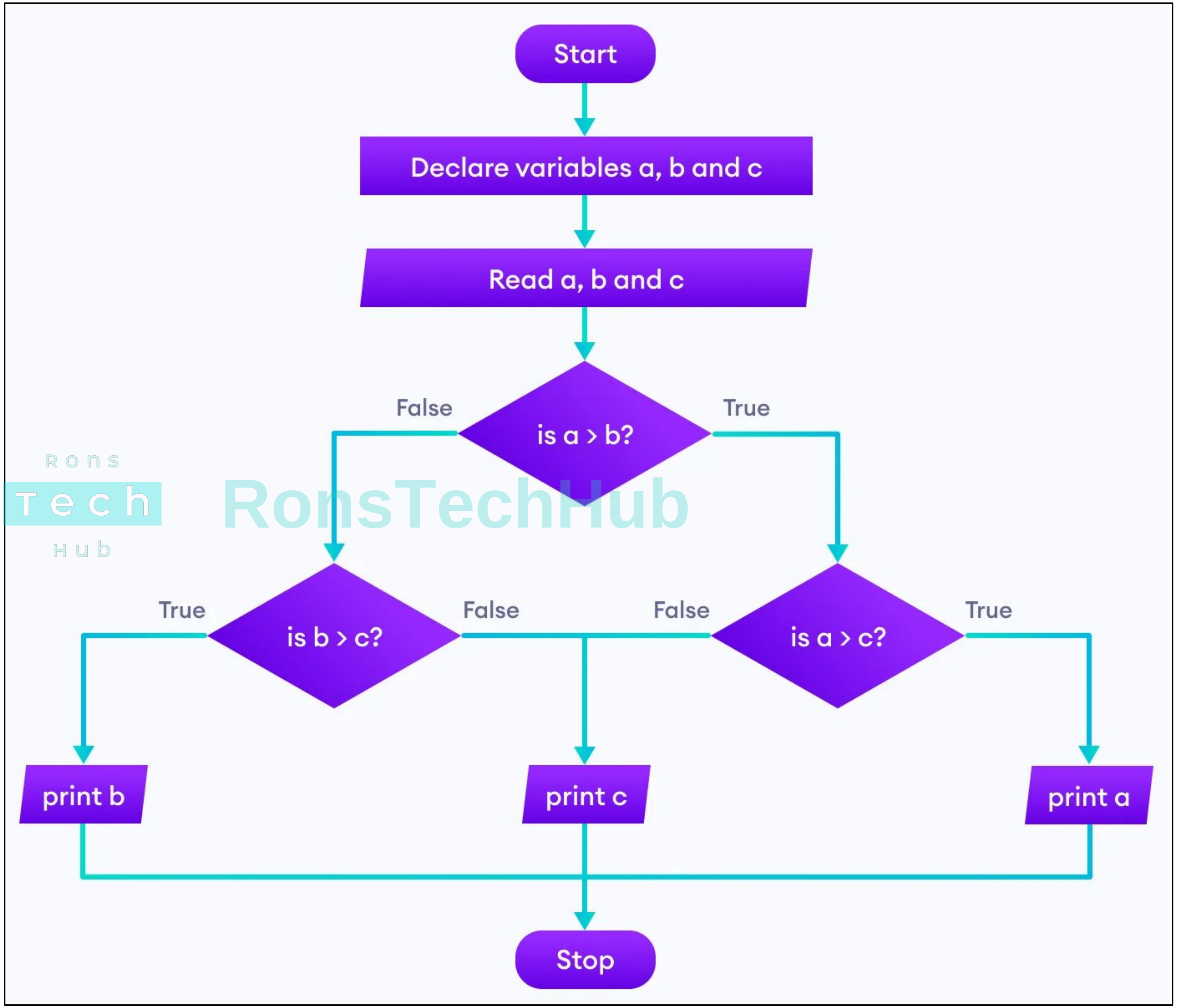
```
REPEAT
    OUTPUT 'What is the best subject you take?'
    INPUT user inputs the best subject they take
    STORE the user's input in the answer variable
    IF answer = 'Computer Science' THEN
        OUTPUT 'Of course it is!'
    ELSE
        OUTPUT 'Try again!'
UNTIL answer = 'Computer Science'
```


- [BBC Bitesize: Pseudocode]

Flowchart Symbols

Name	Symbol	Usage
Start or Stop	Start/Stop	The beginning and end points in the sequence.
Process	Process	An instruction or a command.
Decision	Decision	A decision, either yes or no.
Input or Output	Input/Output	An input is data received by a computer. An output is a signal or data sent from a computer.

Flowchart Example



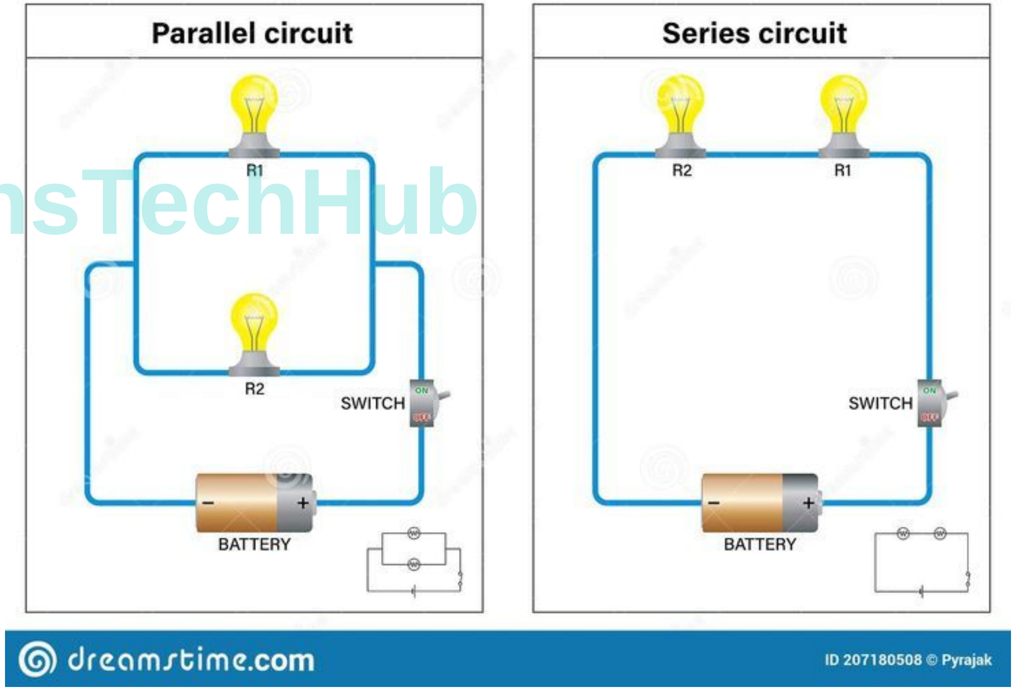
Circuit Diagram

- I will be using Fritzing. I think it is overall the best one to use.

- You can use any program you want.

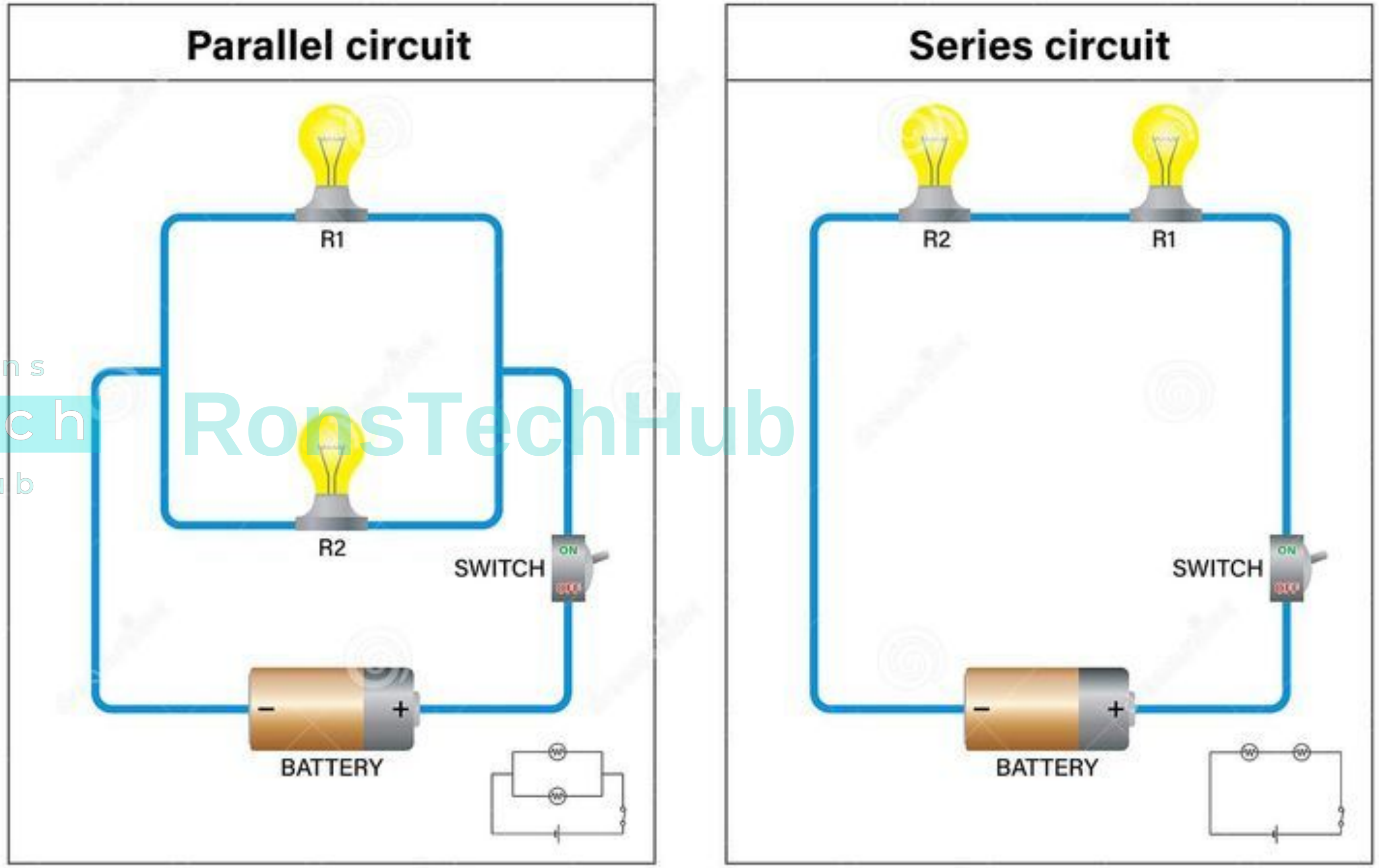
- [Google Search: Best Free Circuit Drawing Software.]

- Ask your school or teacher to look into these.



Circuit Diagram

- [Free Fritzing Alternatives.]
- The best free alternative I have found is the website below. It does Arduinos and Raspberry Pi.
- [WowKi Arduino Simulator.]



Activity 3 Recap

- Parts list with justification.
- Block diagram.
- System/Circuit Diagram.
- Pseudocode OR Flowchart.

Update Activity 1 Log

- I THINK activity 3 might take an entire session depending on the time given.

- You need to go back to your activity 1 log and update all that was done.

- Ensure that the tasks done and the logbook all tie in.

Video On Parts List

Video On Block Diagram

Video On Circuit Diagram

Video On Pseudocode

Rons
Tech
Hub

RonsTechHub

Exam Activity 4

System Assembly and Programming.

This is the implementation section.

You will actually **DO** the thing you planned in system design.

Time: 2.5 Hours.

The Doing

Put	all the pieces together.
Build	your circuit (connect all devices together).
Program	the microcontroller.
Add	comments to each section.

Decompose The Problem

We work based on logic from the Pseudocode and flowcharts.

If possible, build the circuit a section at a time.

This allows you to possibly test each section at a time.

When programming, add a comment to EVERY section or EVER line.

One At A Time

Decomposition Of The Problem

- For each component you have:
  - Build it.
  - Test it.
- This can then be used to not only update the test section.
- It allows you to test each section at a time.
- For example build the LED circuit, test it then put it in a function.

Simplify The Problem

- Build the circuit.
- Program it.
- Test it.
- Repeat until all tasks are complete.
- Add the functions together, the system is complete.

Pseudocode or Flowcharts and Circuit Diagram

Ensure you follow the circuit diagram you designed.

If you needed to make changes update the circuit diagram.

We Built and Programmed

- Now we need to explain what was done.

- Ensure that all your code is commented.

- Even though it is commented I would still explain each major section.

Seems Redundant

- Comments are very short quick bits of information.
- I would do a paragraph for each major section.
- Or bullet points with a lot of detail, I prefer this option.
- Choose the one which suits your style best.
- Ensure you have enough detail.

Rons
Tech
Hub

RonsTechHub

Exam Activity 5

- System Testing and Result Analysis.
- Test Plan followed.
- Results recorded.
- Compare against client brief and requirements.
- Time: 1.5 Hours.

Follow Test and Review Results

Do what you planned to do for the test section, seems obvious.

At the end I THINK you should do an overall comparison.

Your final system versus the client brief/requirements.

This will show how complete your system was based on the requirements.

Update Activity 2

The test plan needs to be updated.

You would have had new ideas of what to test and how to test.

Test based on what the client brief and requirements were.

How and What To Test?

- Make a list of all the things you were supposed to do.
- Detect magnet.
- Keep count of good and bad items.
- If magnet – green LED, increment good count.
- If no magnet – red LED, beep buzzer, increment bad count.

Fill In The Tablet

- Be sure to update the table in activity 5, but with the activity 2 information.

Copy and paste your test plan from **Activity 2** into the table below and complete the **Activity 5** columns.

Activity 2				Activity 5	
Test number	**Purpose of test**	**Test condition**	**Expected result**	**Actual result**	**Comments and justification**
01	Test that LEDs can be triggered properly.	Write program to flash the green LED.	The green LED should turn on for one second and then turn off.	The LEDs were connected to the correct pin and a resistor. They came on as they should.	This was done to test whether or not I would be able to correctly control LED actions. LEDs are to act as a visual representation of the system.
02	Test the buzzer can be activated from the microcontroller.	Write program to activate the buzzer for a single one second buzz.	The buzzer should be audible and stay on for one second.	The buzzer came on when expected.	Like the LEDs the buzzer is to represent the current action of the system but in this case in an audible manner.

Activity 5's Section

Based on the test, you fill in the result you ACTUALLY got.

Comment on why (justification) this test was done.

Analyse The Test Results

- After doing all the tests, you analyse.
- How many things work vs how many do not?
- What were you asked to do vs what you actually got working?

Quick Note

- The exam paper tells you exactly what you have to do for each section.

- No guess work is necessary at all.

From The Exam Paper

- The below points are directly from Activity 5 of the exam paper.

- Test the system using the test plan (from **Activity 2**) and include some unexpected events.
- Record the outcome of each test in the template provided.
- Analyse the test results and evaluate the system for conformance against the client brief.

Unexpected Events

- Power Loss.
- Emergency Stop.
- A sensor error, value not being "0" or "1".
- Off the top of my head.

Unexpected Event – Power Loss

- If you lose power there is not much you can do.
- Maybe a backup generator, in this case a power bank should work fine.
- If you regain power, you can do something.
- On the Pi Pico, name your file "main.py". This will run at start-up.

Unexpected Event – Emergency Stop

This has already been implemented.

I have a start button (button_right) and an emergency stop button (button_left).

Unexpected Event – Sensor Error

- This has been implemented.
- If value == 0, do action 1.
- If value == 1 do action 2.
- Else, do action 3, display error and restart loop.

Test The Unexpected As Well

- Unexpected would indicate it cannot be tested.
- Not true.
- From the list I gave you can test each of those features and note down the results.

Top Tip

Whenever you analyse soemthing you should NORMALLY evaluate as well.

It does not say this on the exam paper.

I would still do this. Link it back to testing.

What Questions To Ask Yourself

Ask	the question.
State	whether or not this was accomplished, yes or no.
Explain	briefly how this was accomplished.

What Questions To Ask Yourself

- Does the system detect a magnet?
- Yes.
- The hall sensor successfully detects whether a magnet is present or not. The value of the hall sensor is a "0" when no magnet has been detected and changes to a "1" when a magnet is present.

What Questions To Ask Yourself

- Does the system count the good and bad items?

- Yes.

- Both the good and bad items are counted. The count for each one is incremented based on whether a condition is met. If the magnet is present the good count gets increased by one each time. If a magnet has not been detected the bad count gets increased by one each time. Only ony count increases at any given time.

What Questions To Ask Yourself

- Does the system display the number of good and bad items?
- Yes.
- The values are displayed to the LCD each time there is a change. This is done roughly every second.

What Questions To Ask Yourself

- Does the system buzz at the correct time?
- Yes.
- If no magnet is detected the buzzer will buzz once then go off.
- If the emergency stop button is pressed the the buzzer will beep every 0.3 seconds.

What Questions To Ask Yourself

- Does the system save information to a text file?
- No.
- When I tried to implement this I ran across a number of issues saving the contens of the text file. I decided it was best to not do this as the inforamtion was already present on the shell and the LCD.

What Questions To Ask Yourself

- Does the system indicate when a batch of 10 is ready?
- I personally forgot about this. In my canse this is a no.
- This is an easy thing to implement so at this stage I would simply go back and fix this.
- *Hint: Look into modulus/remainder division in Python.*

What Questions To Ask Yourself

- Does the system push the good items to one side and the bad items to another?

- No.
- I did not incorporate a motor.
- Like before, this is an easy fix, and I would go back and make the change. Rotate anti-clockwise for bad item, clockwise for good item.

Rons
Tech
Hub

RonsTechHub

Exam Activity 6

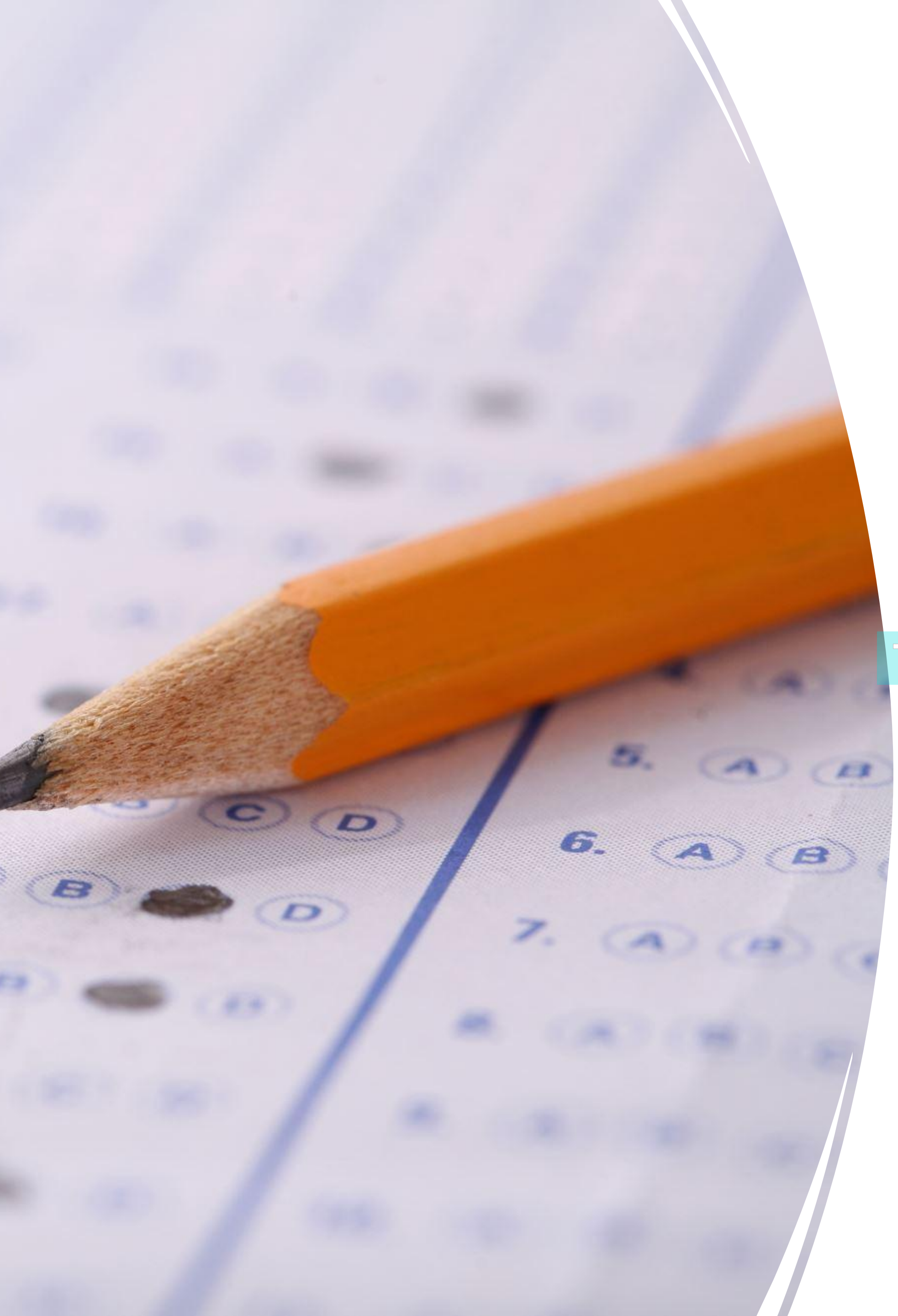
- System In Operation.
- Make a video showing the entire system working.
- Highlight hardware and software.
- Show some tests.
- Time: 2.5 Hours.

Create A Video

That is what needs to be done in activity 6.

DO NOT add anything in the booklet for activity 6.

It is ONLY a video.



The Video Needs To Have...

- Copied from the exam paper.
- Your name, learner registration number and centre number at the start.
- A commentary explaining the operation of the user-friendly system and how its behaviour is linked with your chosen hardware and software program.
- Recorded evidence of the outcome from suitable tests including some unexpected events (from **Activity 5**).

Your Details

- Your name, learner registration number and centre number at the start.
- I would write this on a piece of A4 paper using a black or blue marker.
- Keep it stupid simple and make it easy to read.

What You Should Do/Say In The Video

- A commentary explaining the operation of the user-friendly system and how its behaviour is linked with your chosen hardware and software program.
- Explain how the system works in your own words.
- Speak briefly about the hardware used.
- Maybe list the hardware (Pi Pico, 2 LEDs, LCD, Buzzer etc.)

What You Should Do/Say In The Video

- Speak about the software.

- How did you program it.
- I used Functions, While Loops and If Statements.

What You Should Do/Say In The Video

- When started the program does a countdown from 3 to 1, then indicates it is ready to start.
- If the start button is pressed, the main function is run.
- The main function checks if there is a magnet is present.
- If a magnet is present (green led, increment good count, push item right).
- If no magnet (red led, increment bad count, buzz, push item left).

What
You Should Do/Say
In The Video

- If the stop button is pressed.
- The system will pause, flash all the LEDs and buzz continuously, until the start button is then pressed.
- If the system losses and regains power, the program will start automatically as the Python file is named, "main.py".
- A feature implemented by the Raspberry Pi designers.

Show Tests

- Recorded evidence of the outcome from suitable tests including some unexpected events (from **Activity 5**).

- Test each thing you mentioned in Activity 5.
- Also show the unexpected events.
- The ones I chose were:
  - Power loss. Solution: main.py
  - Emergency Stop. Solution: Emergency Button
  - Sensor Error. Solution: Else to catch anything out of the ordinary.

Rons
Tech
Hub

RonsTechHub

Admin Stuff

You need to rename your files as described on the Exam Paper.

You need to export your document as a PDF.

Authentication sheet.

Naming Files

- This information is on the Exam Paper.

Each learner will need to submit evidence using the file names below:

- Electronic task booklet: booklet_[Registration number #]_[surname]_[first letter of first name]
- Audiovisual file: file_[Registration number #]_[surname]_[first letter of first name]

Document Name

booklet_[Registration number #]_[surname]_[first letter of first name].

Booklet_909090_Kingboss_R

Video Name

- file_[Registration number #]_[surname]_[first letter of first name].
- File_909090_KingBoss_R.