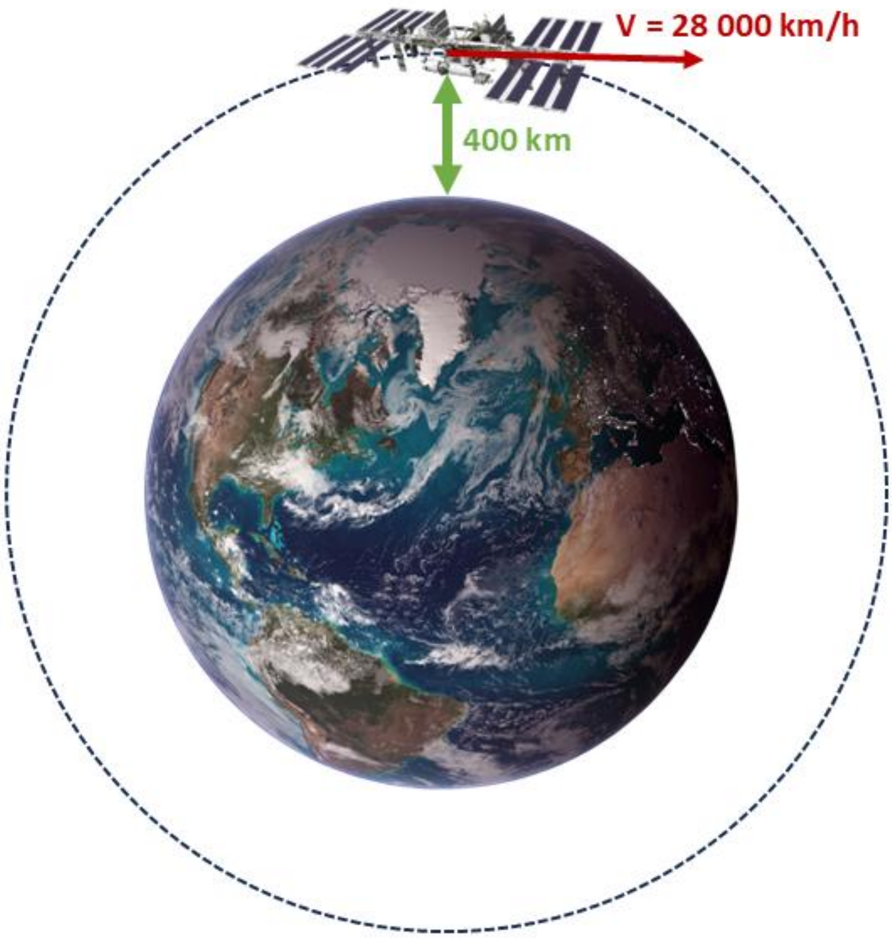
Calculating the speed of the ISS

Program your Astro Pi

Activity 1: The International Space Station

The International Space Station (ISS) was established at the end of the last millennium through a collaboration between the European, Japanese, Canadian, Russian and US space agencies (with the latter two being the main investors). In particular, it succeeded the Soviet MIR space station. Orbiting at an altitude of approximately 400 km, the ISS is the largest satellite ever built. Given its average speed of 28,000 km/h and its proximity to Earth, the ISS orbits our planet 16 times a day. Astronauts on board can therefore witness up to 16 sunrises and sunsets a day.
The main objective of the ISS is to study the effects of microgravity. To this end, numerous experiments are carried out continuously and simultaneously on board to study cardiovascular system function, analyze the properties of specific fluids and materials, and observe plant growth.
Raspberry Pi computers have been on board the ISS since 2015, with the first units installed by astronaut Tim Peake. As part of the Astro Pi project, they enable students to conduct scientific research in space by writing short computer programs that can be run on these on-board computers.

British astronaut Tim Peake holding an Astro Pi on board the ISS © ESA



A few questions about the International Space Station:

What is the mass of the ISS?

The mass of the ISS is approximately 420 tonnes.

In which year was the space station's first module launched? What is its name?

The first module of the space station, the Russian Zarya module, was launched in 1998.

What is the size of the space station?

The ISS is 109 meters long and 73 meters wide, roughly the size of a football pitch.

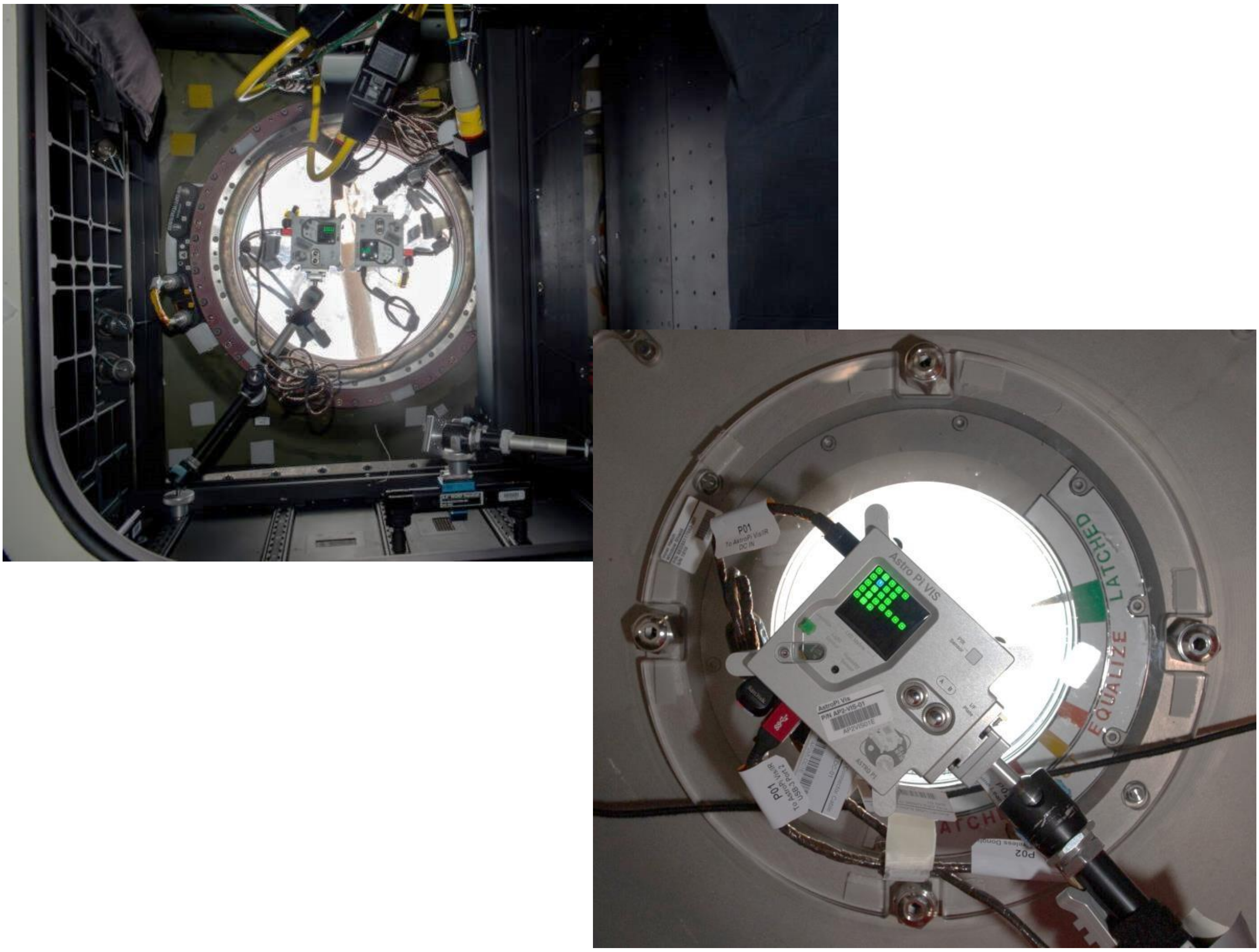
On average, how many astronauts are on board the ISS?

On average, seven astronauts are on board the ISS

Activity 2: Calculate the speed of the space station

A. The Astro Pi computers

The two Astro Pi cameras aboard the International Space Station are positioned facing a window overlooking Earth. The cameras can thus be triggered by a program written in Python, which we are going to create. It is even possible to annotate the photos to record their location as well as thedate and time they were taken.



© ESA

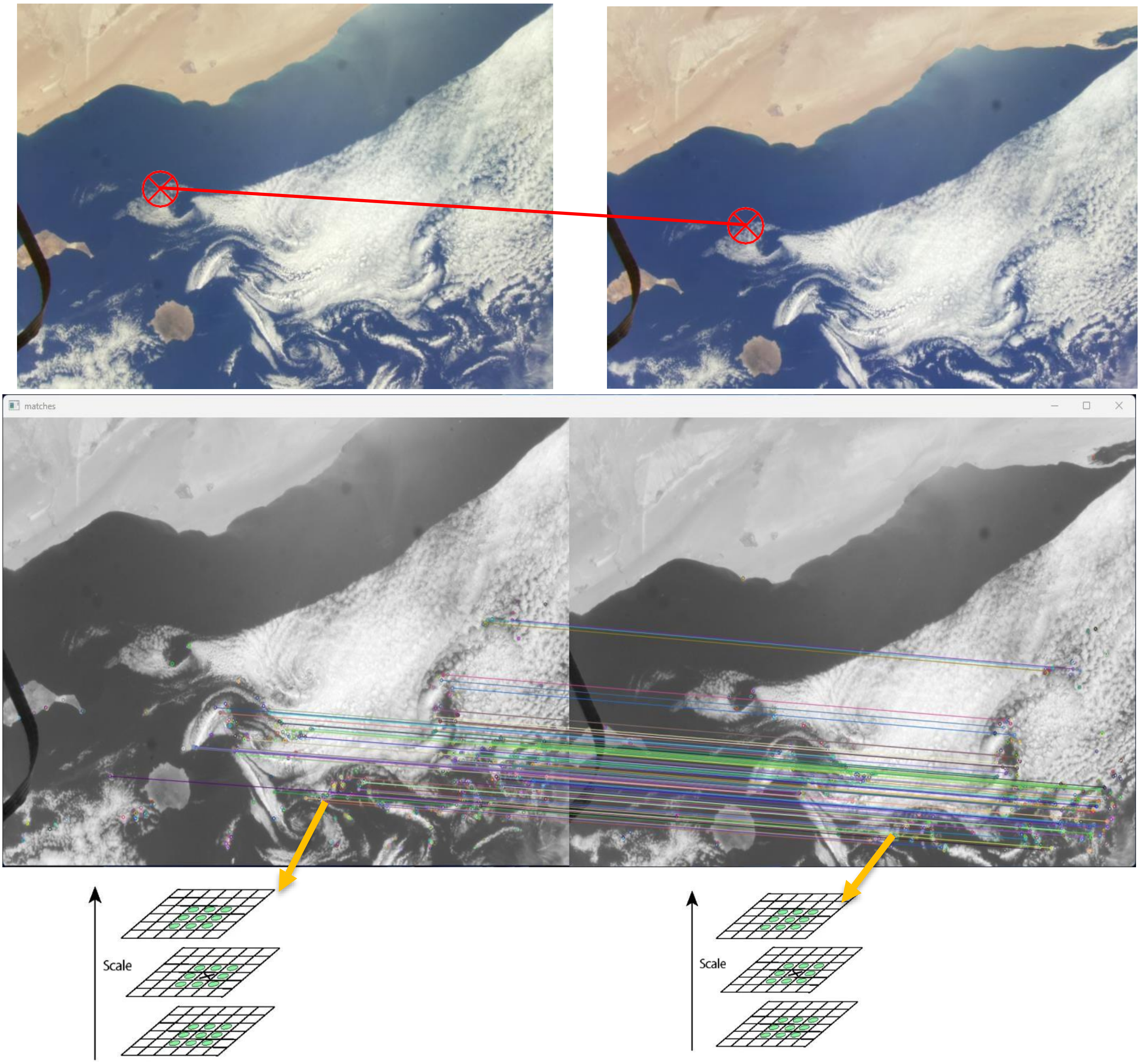
B. Calculating the speed of the ISS using an Astro Pi

, The speed of the space station can be determined using photos taken with Astro Pi cameras. To do this, simply take two photos a few seconds apart. By knowing the image scale and tracking the movement of two specific points between the photos, the distance traveled over a given time can be calculated.

This can be achieved using the [OpenCV] software and the [SIFT] (Scale-Invariant Feature Transform) method. This software uses artificial intelligence to locate "descriptors" within the images—analyzing images 1 and 2 point by point to identify "matches" or common features.. The software can then determine the distance between these two points and automatically calculate the speed using the following formula:

$$\boldsymbol{Speed} = \frac{\boldsymbol{Distance\ covered}}{\boldsymbol{Travel\ time}}$$

Picture 1: taken at t0 = 0 seconds **Picture 2: taken at t1 = 5 seconds**

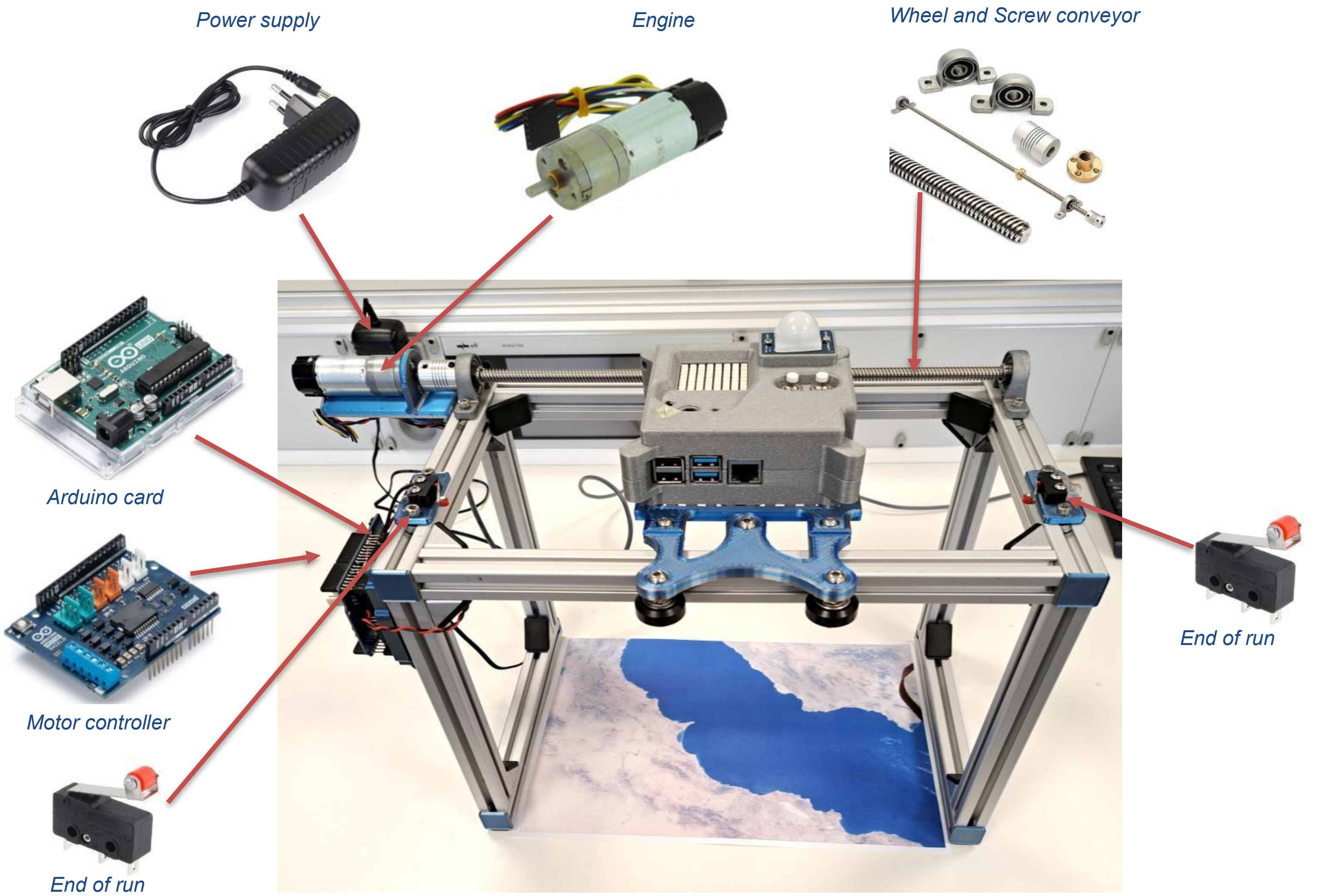


© ESA

C. Introduction to the test bench

To develop and design your program, a test bench allowing you to simulate the movement of the ISS is available [here](#).

To do this, the Astro Pi unit is installed on a motorized trolley that allows it to move linearly. The camera is positioned above a photograph of Earth taken from the ISS. The trolley will therefore move continuously from left to right, enabling the camera to capture images while in motion, just as it would aboard the ISS.



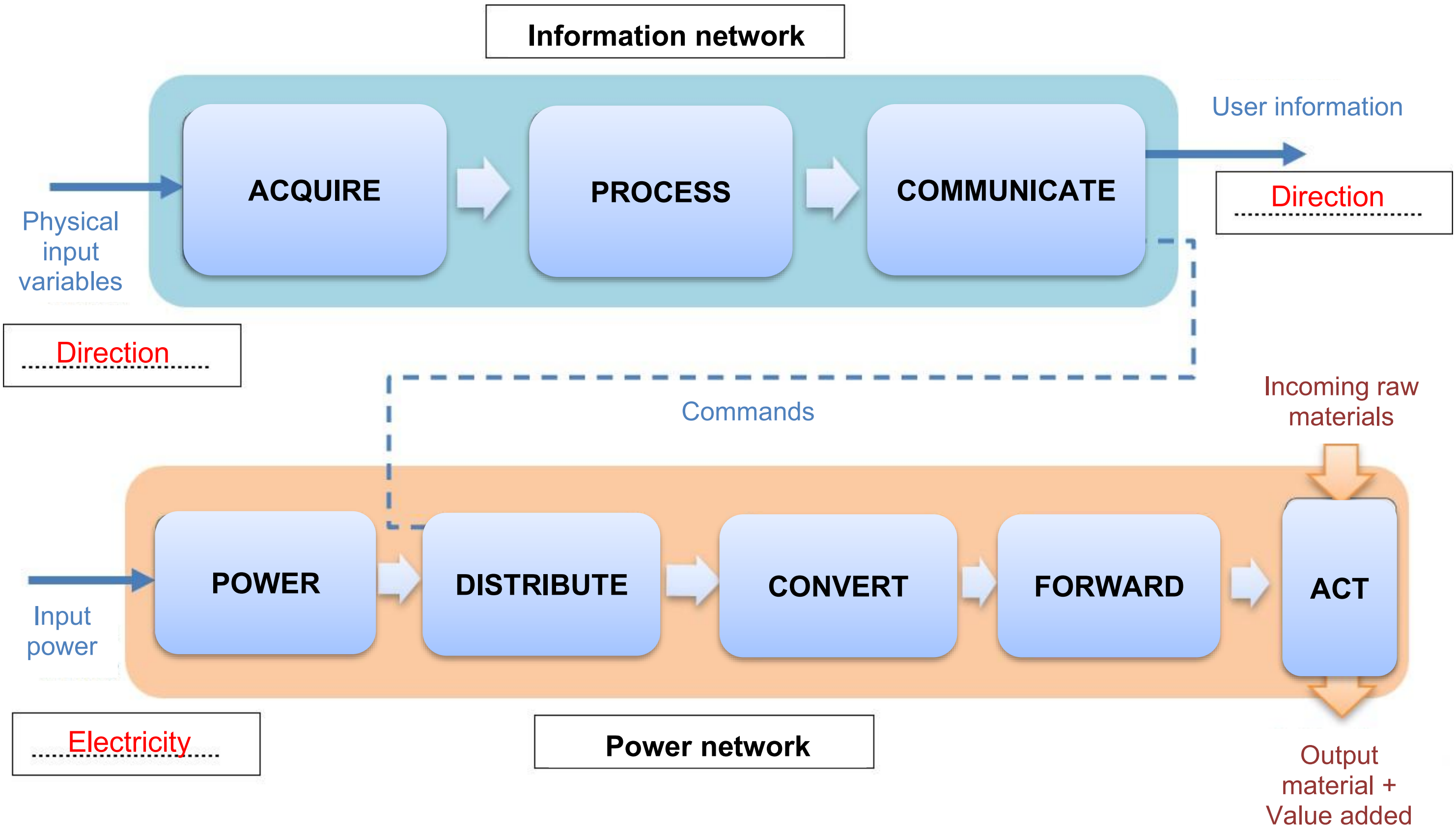
a. Test bench analysis

1. **Complete the following table using the 'energy chain/power chain' diagram. For each element, indicate the nature of the functional block: acquisition, processing, communication, etc.**

Items	Description	Functional block
	**Limit sensor:** These two sensors indicate whether the trolley has reached its maximum left or right position.	Communication
	**Screw conveyor system:** Transmits motion from the motor to the trolley, converting rotational motion into linear motion.	Transmission

	**Arduino Card**: Enables the system to be controlled by a program that processes incoming data and issues commands to the motor based on the information collected.	Processing
	**Motor control board:** Enables power to be transmitted to the motor so that it can rotate in either direction as required.	Distribution
	**Power supply unit:** Supplies electrical power to the system.	Power supply
	**Engine:** Converts electrical energy into mechanical rotational energy.	Conversion

2. Complete the information chain and power chain diagram



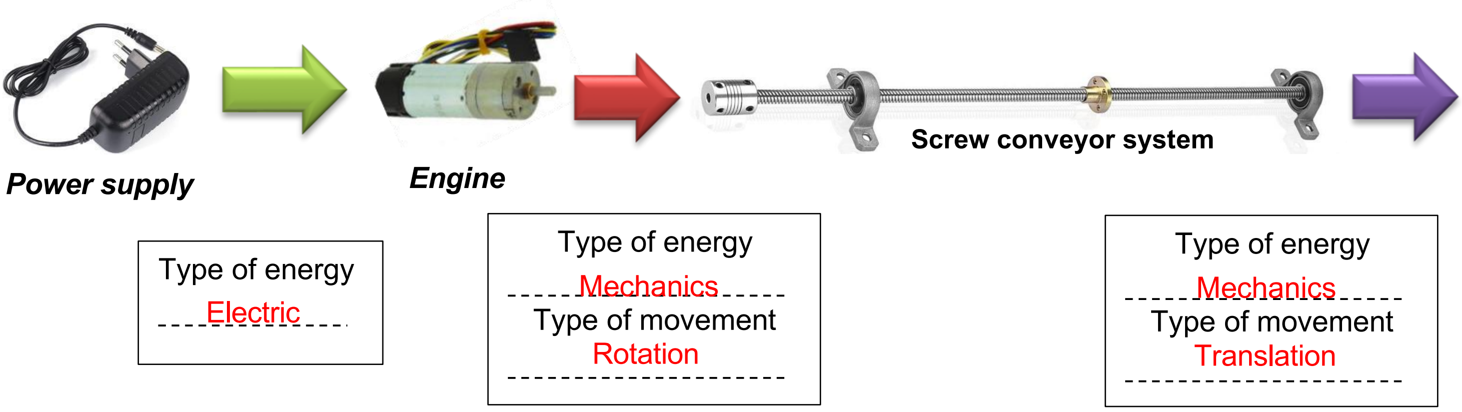
- Incoming raw materials:

  Electrical power and sensor signals

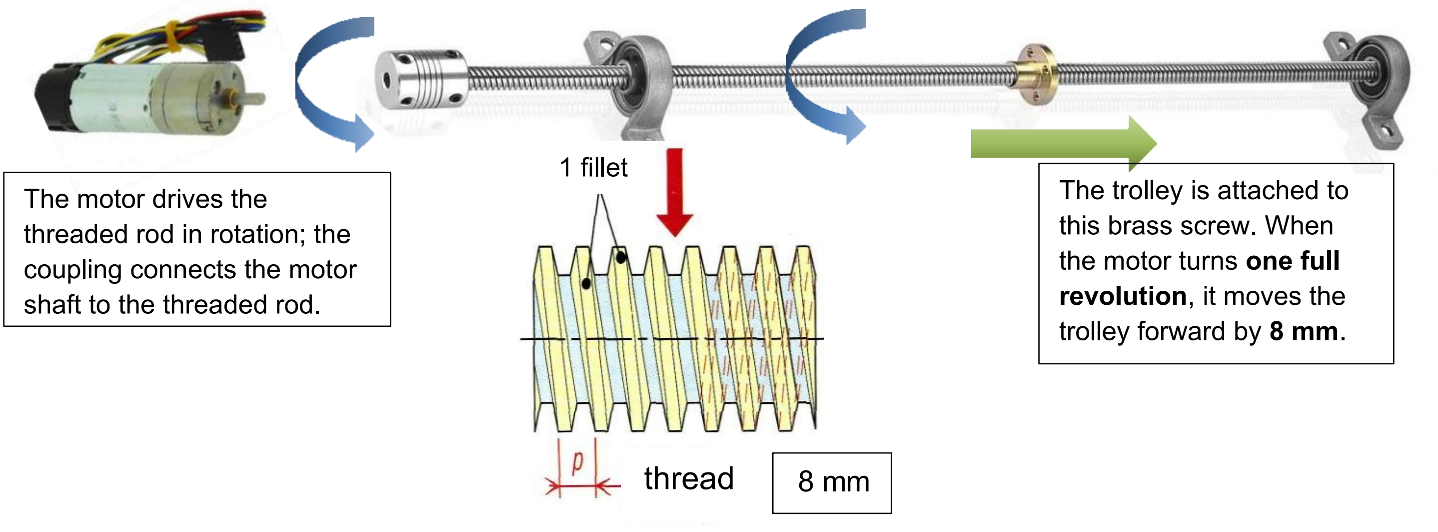
- Output material + Added value:

  Outgoing material: Precise trolley movement, controlled by the system
  Value added: Controlled linear movement to position the trolley at one end or in the center

- Power chain: Complete the following diagram by indicating the type of energy (mechanical or electrical) that is transferred between the components, and the nature of the movement between them (rotation or translation).



3. Determine the travel speed of the trolley



Given that the motor runs at a speed of 200 rpm, calculate the speed of the trolley in m/s and then in km/h:

Calculation details:

$$\text{Motor rotation speed} \times \text{trolley displacement per revolution} = \text{Motor travel speed}$$

Given that there are 60 minutes in an hour, 60 seconds in a minute, and therefore 3,600 seconds in an hour.

$$200 \times 8 = 1\,600\ mm/min$$

$$1600 \div 1000 = 1{,}6\ m/min$$

$$1{,}6 \div 60 = 0{,}027\ m/s$$

$$0{,}027 \div 1000 = 0{,}000096\ km/min$$

$$0{,}000096 \times 3600 = 0{,}35\ km/h$$

Results :

Trolley speed = 0,027 .... m/s

Trolley speed = 0,35 .... km/h

b. Moving the trolley

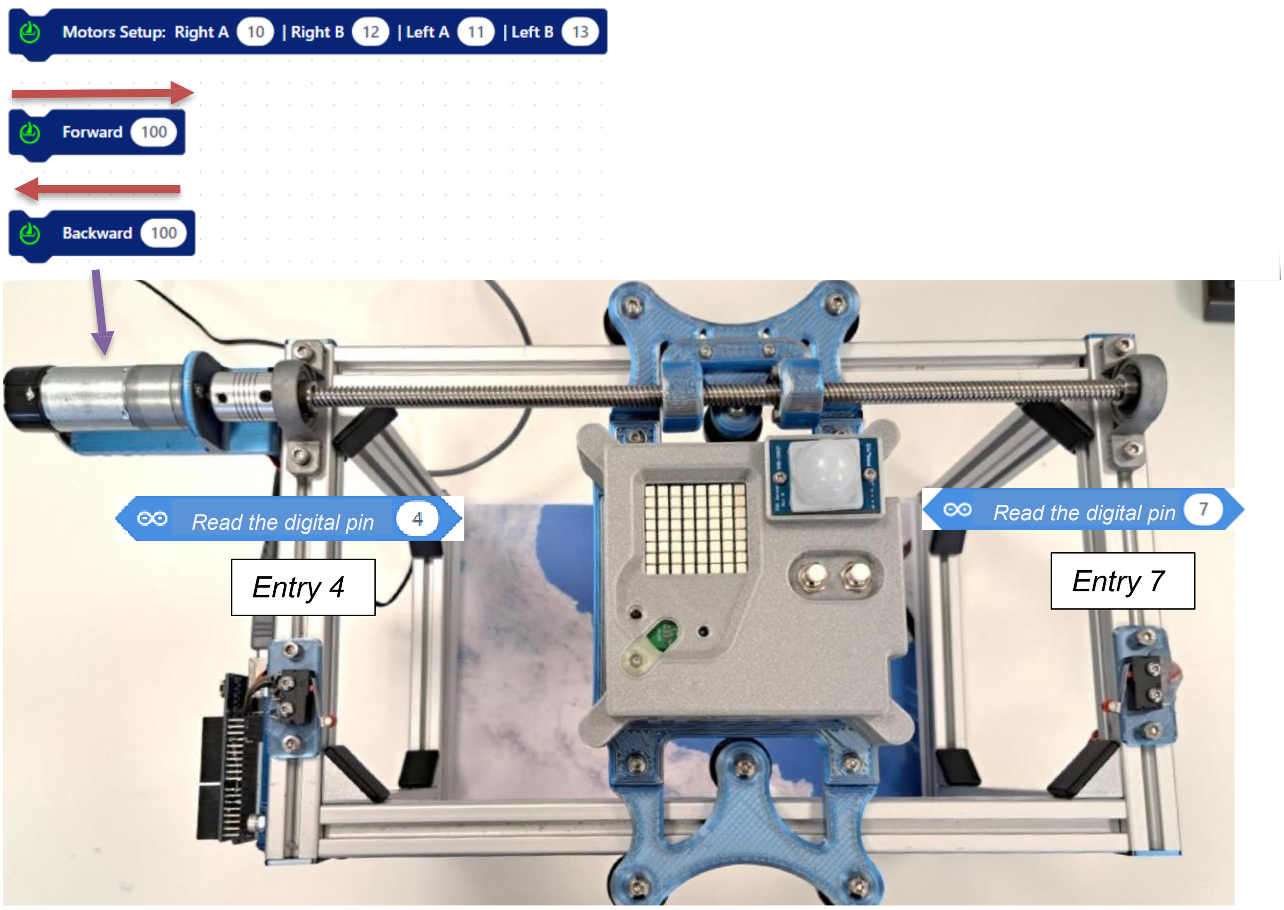
1. Before starting

- Launch the Mblock 5 software
- Go to the 'Device' menu
- Upload the Arduino UNO card
- Go to the 'Extension' menu, which is located below all the blocks
- Search for the SYSria L298P EN extension and add it.

2. Make a moving program

Design an algorithm to move the trolley continuously. To ensure it moves from left to right and then from right to left without stopping, the motor's rotation direction must be reversed when the trolley touches the left or right limit sensor. The program must run for 10 minutes and then stop.

3. Submit your program to your teacher for testing



D. Taking a photo with Astro Pi

Create a program that takes a picture and saves it to the Astro Pi computer

The Astro Pi unit is programmed using Python. To simplify writing the program, you can use the Mblock software to create the program using blocks, and then convert it into Python code.

Before starting, set up the Mblock software

1. Launch **Mblock 5.**
2. On the left side of the menu, click '**Add**' and load the Raspberry Pi card from the '**Device**' menu.
3. Click on the '**Extensions**' button and install the '**Sense Hat**' and '**Pi Camera**' modules.

Program to be executed: take a picture automatically every 30 seconds in a loop (the program runs in an infinite loop).

- The camera used in the Astro Pi unit is the HQ camera. Using the resource document, state the maximum resolution of this camera.

The resolution of the HQ camera is 12.3 megapixels

- Using the 'Camera' blocks, try to create a program that takes a picture at the camera's maximum resolution and saves it with a name such as picture 1, picture 2, etc., without overwriting the previous picture file. The program must run for 10 minutes.

Present your program

Hardware Specification

	Camera Module v1	Camera Module v2	Camera Module 3	Camera Module 3 Wide	HQ Camera	GS Camera
Net price	$25	$25	$25	$35	$50	$50
Size	Around 25 × 24 × 9 mm	Around 25 × 24 × 9 mm	Around 25 × 24 × 11.5 mm	Around 25 × 24 × 12.4 mm	38 x 38 x 18.4mm (excluding lens)	38 x 38 x 19.8mm (29.5mm with adaptor and dust cap)
Weight	3g	3g	4g	4g	30.4g	34g (41g with adaptor and dust cap)
Still resolution	5 Megapixels	8 Megapixels	11.9 Megapixels	11.9 Megapixels	12.3 Megapixels	1.58 Megapixels
Video modes	1080p30, 720p60 and 640 × 480p60/90	1080p47, 1640 × 1232p41 and 640 × 480p206	2304 × 1296p56, 2304 × 1296p30 HDR, 1536 × 864p120	2304 × 1296p56, 2304 × 1296p30 HDR, 1536 × 864p120	2028 × 1080p50, 2028 × 1520p40 and 1332 × 990p120	1456 x 1088p60
Sensor	OmniVision OV5647	Sony IMX219	Sony IMX708	Sony IMX708	Sony IMX477	Sony IMX296
Sensor resolution	2592 × 1944 pixels	3280 × 2464 pixels	4608 x 2592 pixels	4608 x 2592 pixels	4056 x 3040 pixels	1456 x 1088 pixels

Finally, add the section of code featuring artificial intelligence to complete the program.