{Learn, Create, Innovate};

Challenges

Mini challenge 3

MCR²

Manchester **Robotics**

Mini Challenge 3

Introduction

This mini-challenge is intended for the student to review the concepts introduced in the previous sessions.

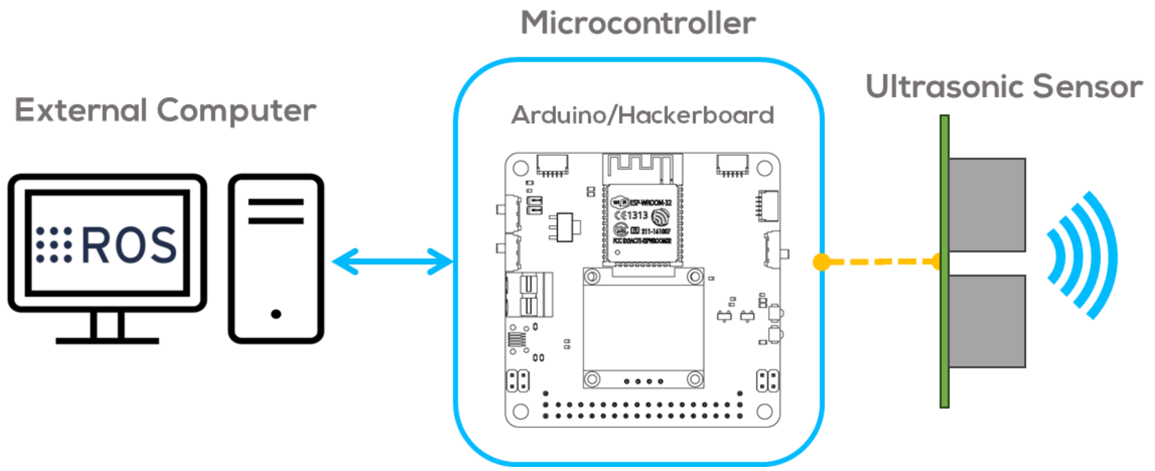
- The robot to be used in the following sessions by the students are the EAI Smart Robots.
    - The robot contains several ultrasonic sensor modules.
- Such sensors are typically used for obstacle avoidance or other “reactive” navigation techniques.

Mini Challenge 3

Introduction

These sensors are typically connected to low-level control devices in a master-slave architecture, such as microcontrollers or other embedded solutions; to save processing time and allow the computer to focus on more complex tasks.

- The activity involves creating a ROS node to communicate an ultrasonic sensor with the computer using ROS.
- The sensor will be read using a microcontroller (Arduino Board) as an intermediary to save the computer's processing time.
  - See the following slides for requirements.



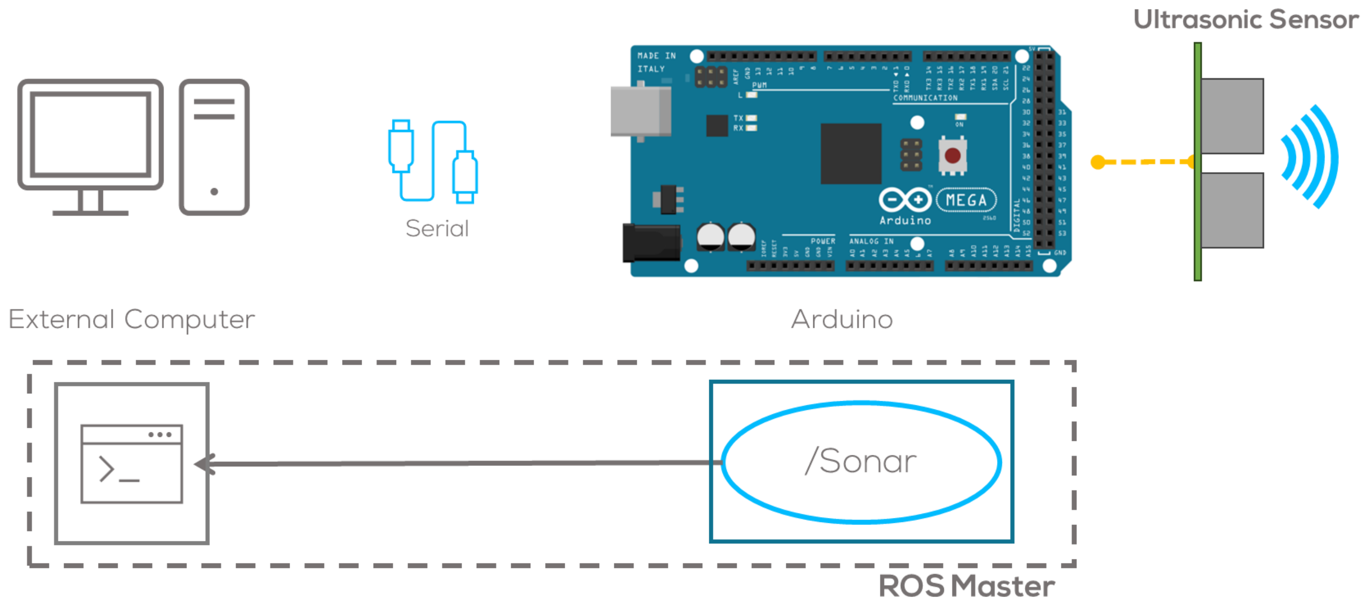
Mini Challenge 3

Challenge

The student must be able to visualize the data provided by an ultrasonic sensor using ROS.

To do this, the student must program a node that publishes the data obtained by the microcontroller from the ultrasonic sensor.

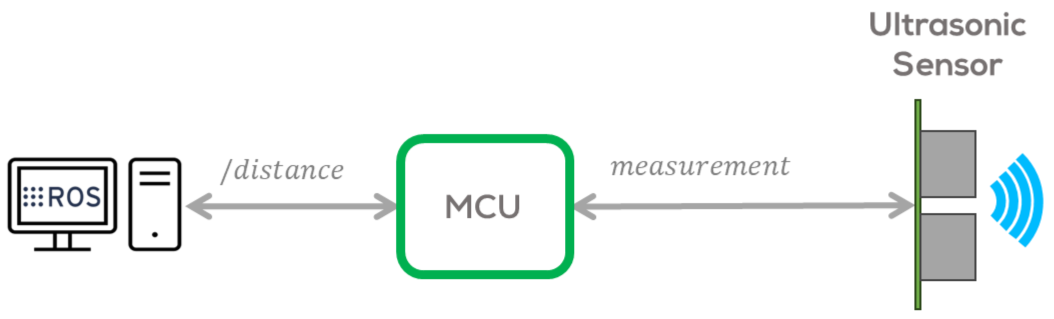
- Connect a sonar to the Arduino module and publish the data range of the sonar using ROS.
- The student must develop the communication libraries for the communication between the sonar and the Arduino. More information about the manufacturer's protocol here.
- The communication must be performed via serial, using the rosserial library for Arduino.

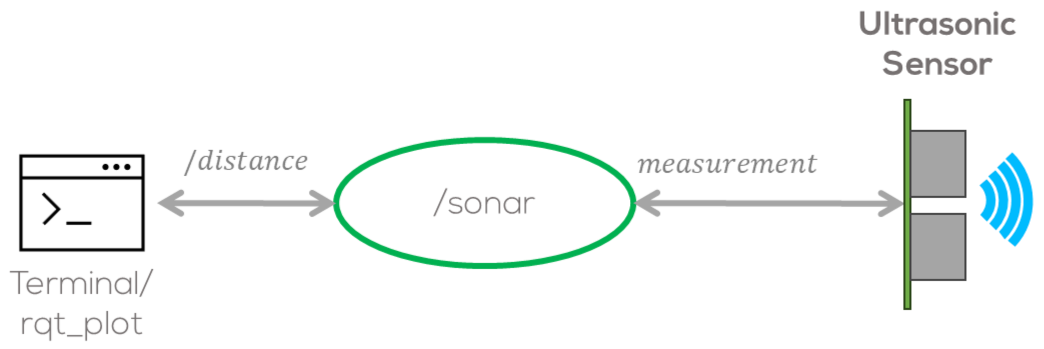


Mini Challenge 3

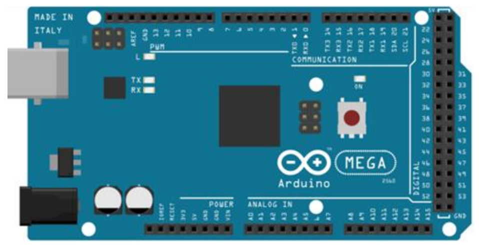
Challenge

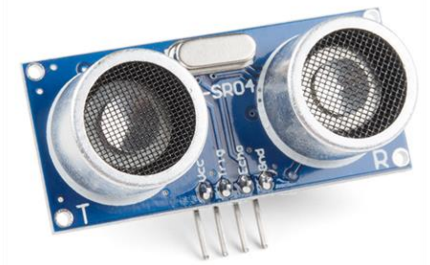
- The information from the sensor must be published using the topic "/distance"
- The measurement must be published using a Float32 message.
- The result must be plotted using any visualizer available such as: "rqt_plot".
- The nodes and topics must be visualized using the "rqt_graph".



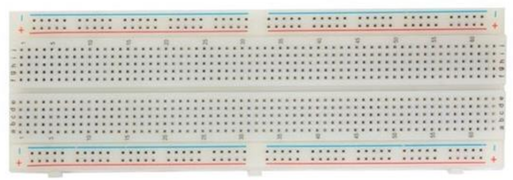


Mini Challenge 3: Requirements

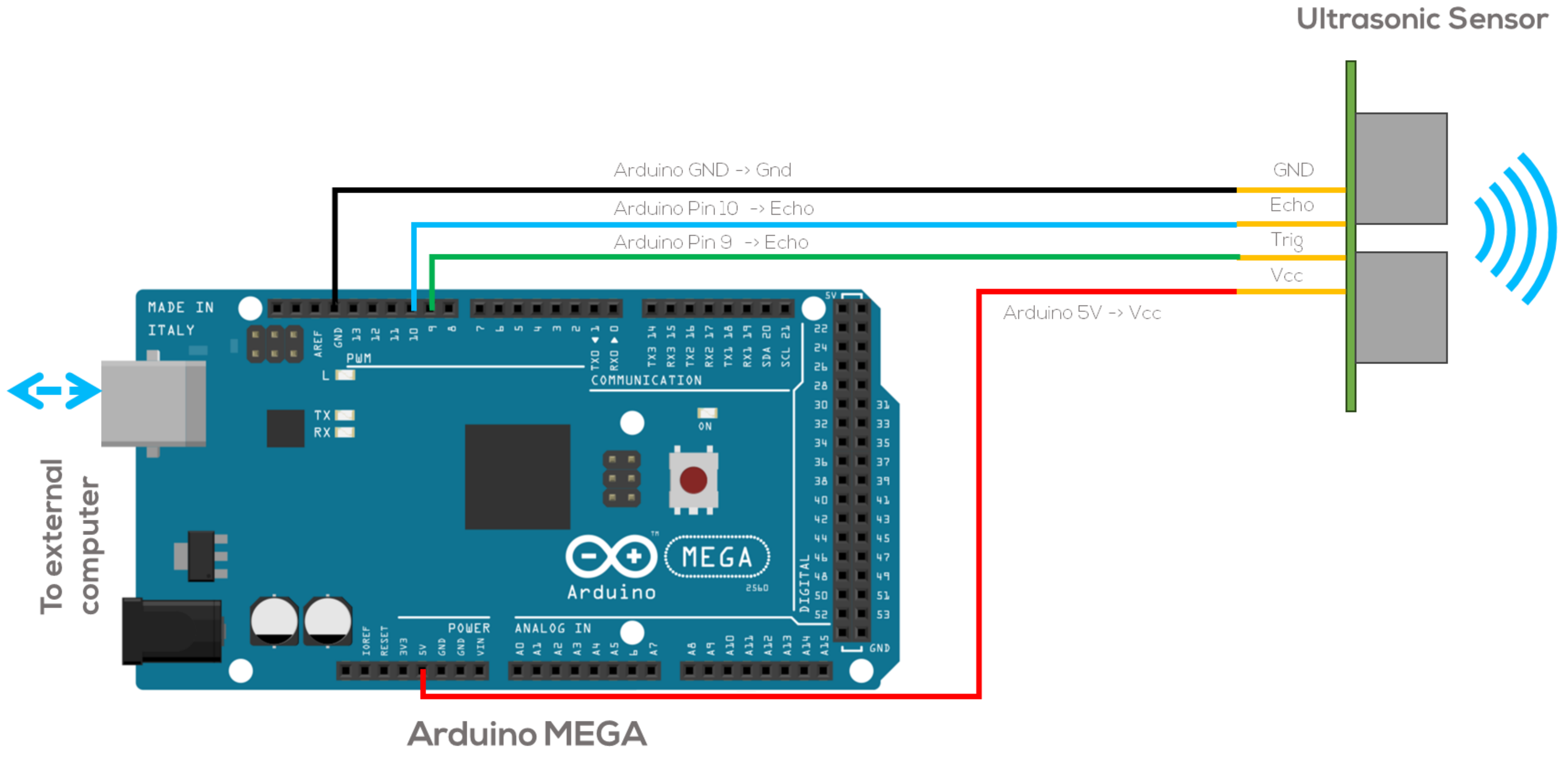

Arduino MEGA


Ultrasonic Module HC-SR04 (or similar)

Wires (Dupont or any wire)


Breadboard

Mini Challenge 3: Connection Diagrams



Deliverables

Teams:

- The students must form teams for this mini-challenge. The teams will be the same as in other classes of this concentration.
- Th teams must be multidisciplinary.
- The students must respectfully help each other to understand all the topics.
- The team must manage the project, using a project management methodology, and present it in the report.
  - The methodology selected can be simple, E.g., Waterfall, Agile, Kanban, etc.

REPORT

- Maximum number of pages: 2 pages, not including presentation.
- Presentation: Include name, student ID, and team name.
- Minimum font size: 11 pt.
- Appendix: No.
- Report type: Individual.
- Report design: Single-column or two columns.
- Format: PDF
- Content Details:
  - Each task must be included in a different section.
  - It is recommended to use diagrams, figures, tables, etc.
  - Include reflections, conclusions, y recommendations of each result in detail.
  - Include references using the IEEE format.
- Important: Results and/or figures that do not contain information or detailed explanations will be penalized.

Deliverables

Rubric

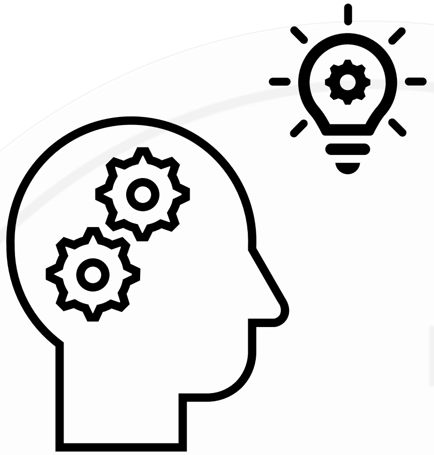
- Introduction
  - Explain the problem and why is essential.
- Explain the project management strategy selected (use diagrams, flow chart)
  - The student must show diagrams explaining how the project management strategy was implemented.
  - The student must describe concretely, why this strategy was selected.
- Explain the code, and how it works? Constraints? Advantages, and disadvantages? etc.)
  - The student must explain the code using diagrams, flowcharts, etc.
  - The student must show the proposed solutions' constraints, advantages, and disadvantages.
- Plots used to verify and analyse the behaviour of the algorithms, showing the performance of the sensor
  - The student must show the plots used to verify the behaviour of the algorithm estimated robot position over time, control input, etc.
- Reflect on the sensor's performance and propose solutions (What problems can be expected? Solutions/improvements?)
  - The student must provide a small reflection on the problems that can occur using their algorithm with ROS.
  - *The student is expected to explain the physical phenomena that make obtaining measurements difficult, if applicable.*
  - The student must propose some solutions on how to solve these issues.
- Report presentation, references, and clarity
  - Report must be clear for the reader, well organised, and with a good presentation and references.

Mini Challenge 3: Extension 1 (extra marks, not mandatory)

Be creative!

Generate a node on the computer that subscribes to the "/distance" topic and use the information of the sensor to perform a simple task in ROS, e.g., Distance alarm, parking sensor, etc.

- Must be implemented using ROS.
- The report must clearly define and describe the task (1/2 a page to 1 page maximum added to the report).
- Usage of diagrams, images flowcharts is encouraged.
- A video (YouTube) showing how it works must be presented.



Mini Challenge 3: Extension 2 (not mandatory, no extra marks)

Challenge Extension 2

Publish the measurement using an appropriate standardized message for range measurements in ROS.

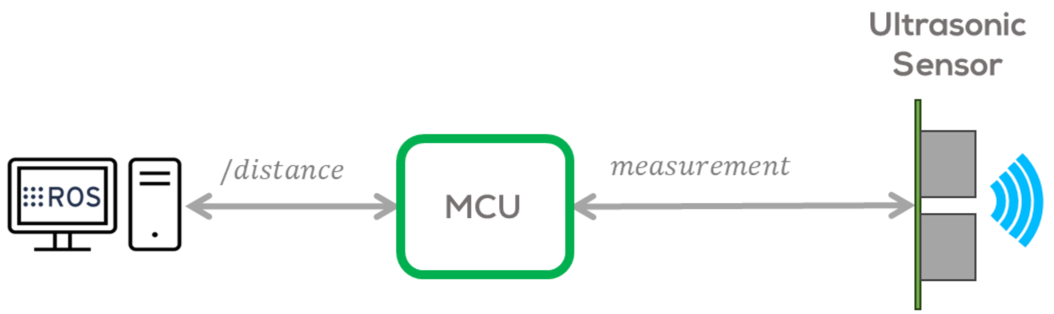
- "Range" is the typical standard message in the "sensor_msgs" library. More information here.
- Hint 1: call the library on the Arduino as follows "*#include <sensor_msgs/Range.h>*"
- Hint 2: Do not forget the time stamp of the message, you can define it as follows,

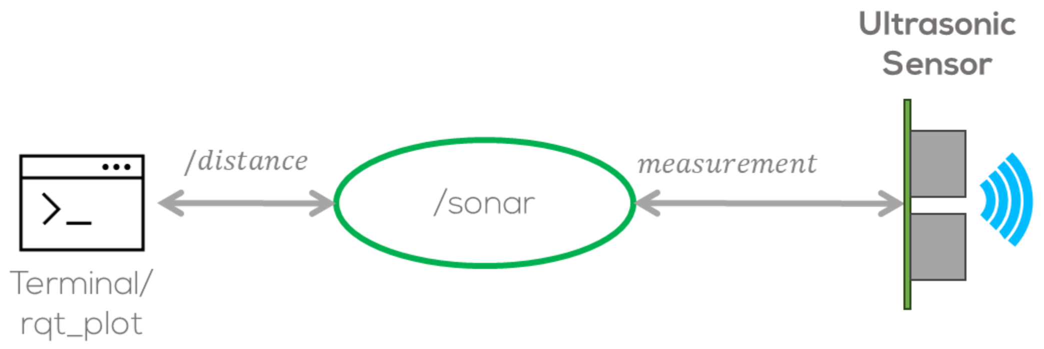
  "message_name.header.stamp = nh.now();"

  - Where nh.now() is the ROS time inside the library

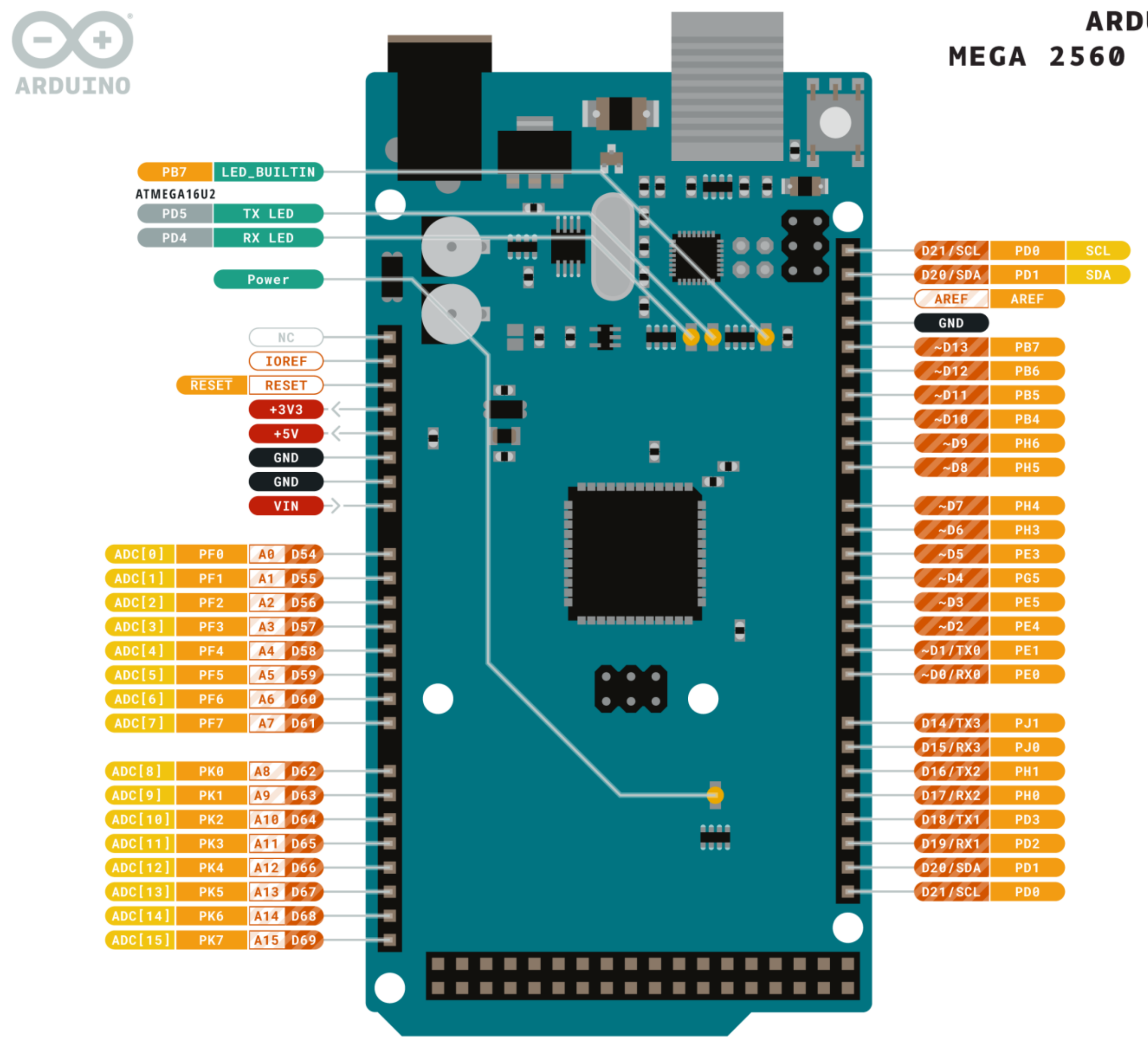
    "#include <ros/time.h>"

- The results of this task must be included in the report in less than 1/2 page.





ARDUINO MEGA 2560 REV3



■ Ground	■ Internal Pin	▨ Digital Pin	■ Microcontroller's Port
■ Power	■ SWD Pin	▨ Analog Pin	
■ LED	□ Other Pin	■ Default	

ARDUINO.CC

Rules

- This is challenge, **not** a class. The students are encouraged to research, improve tune explain their algorithms by themselves.
- MCR2(Manchester Robotics) Reserves the right to answer a question if it is determined that the question contains partially or an answer.
- The students are welcome to ask only about the theoretical aspect of the class.
- No remote control or any other form of human interaction with the simulator or ROS is allowed (except at the start when launching the files).
- It is **forbidden** to use any other internet libraries except for standard libraries such as NumPy.
- If in doubt about libraries, please ask any teaching assistant.
- Improvements to the algorithms are encouraged and may be used if the students provide the reasons and a detailed explanation of the improvements.
- All the students must respect each other and abide by the previously defined rules.
- Manchester Robotics reserves the right to provide any form of grading. Grading and grading methodology are done by the professor in charge of the unit.