**AL5D Robot arm project.**

SHOULD WE MAKE THE CONTACT PLATE OURSELVES OUT OF COPPER?

# Problems to be solved

* Implement kinematic algorithm to move robot arm to location of resistor to be sorted.
* Read the value of a resistor using the 2 probes on the modified gripper design.
* Turn electromagnet on.
* Based on the returned value, tell the robot arm to sort the resistor into the appropriate container.
* Turn electromagnet off.
* Implement safety features (master stop).
* Implement Calibration features.
* Possibility of datalogging.

# Technical considerations.

Due to the reliability of serial communication over the working envelope of the robot arm:

(<https://www.digikey.com/en/articles/uarts-ensure-reliable-long-haul-industrial-communications#:~:text=With%20appropriate%20line%20drivers%2C%20a,485%20or%20RS%2D422%20interfaces>.)

*“With appropriate line drivers, a UART can work over long distances: from 15 meters (m) for the RS-232 serial data bus to 1000 m for RS-485 or RS-422 interfaces” [22 May 2019]*

The proposed design utilises an ohm meter on the arm itself, to read the value of the resistor and increase the reliability of that analogue reading, then send the processed reading to another “master” microcontroller which can then activate the robot arm. This will mean making 2 PCBs, one which could be mounted on the gripper and one to remain at the base acting as a master controller.

Diagram

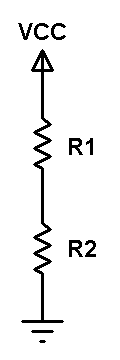
Description automatically generated

# Ohm Meter

Given below is the theory behind the ohm meters analogue peripheral to provide an auto ranging feature.

Utilising the ADC peripheral on the ATMEGA328p, resistance could be read from an unknown resistor by creating a voltage divider with that and another resistor of known value. Reading the voltage from that divider will give the resistance.

The resistance of known value is not random and does need to be within a relative range of that which is being tested. Thus, a feature which is seen in most multimeters, an auto ranging feature must be implemented to allow a wide range of values to be read without manually changing the resistor of known value.



Applying voltage divider equation to the above circuit we get:  
Voltage across resistor R1: VR1 = VCC x R1/(R1 + R2)  
Voltage across resistor R2: VR2 = VCC x R2/(R1 + R2)

Assume that the resistance of R1 is known and R2 is unknown, by applying voltage divider equation we can easily get the value of R2 by measuring the voltage across it where: R2 = VR2 x R1/(VCC – VR2). If VR2 is equal to VCC then R2 = infinite.

**How to measure R2 voltage (VR2):**  
The voltage is an analog signal and we can measure it using ADC (Analog-to-Digital) converter which converts analog data into digital data. The Arduino board with ATmega328P microcontroller (UNO, Nano, Mini …) has a built-in 10-bit ADC module where VCC (+5V) is represented by 1023 and 0 volts is represented by 0 (if VDD is already used as positive voltage reference).

**Do we need fixed voltage reference?**  
Actually, I think there’s no need for a fixed voltage reference and adding it will not give a big change in the results. Because the voltage divider positive terminal (VCC) and the ADC positive reference voltage are connected to the same point and any change of this point voltage will affect both the voltage divider output and the ADC module.

**Multirange ohmmeter:**  
To get a good value of the resistor to be measured, a multirange ohmmeter should be used.  
The multi-range ohmmeter may also use voltage divider technique with different values of resistor R1. The following image shown a simple diagram of multi-range ohmmeter:

**[Diagram, schematic

Description automatically generated](https://simple-circuit.com/wp-content/uploads/2020/01/multirange-ohmmeter.png)**

The multi-range ohmmeter is based on a mechanical switching device with 1 input and multi output. At any time just one known resistor (R11, R12 … R1n) is connected in series with the unknown resistor R2.

**Auto-ranging ohmmeter:**  
The auto-range ohmmeter operates the same as the multi-range ohmmeter except that the mechanical switching device is replaced by electronic switches such as transistors driven by an intelligent device (microprocessor, microcontroller …) that decides which switch is closed according to the resistance value of the unknown resistor.

Project Hardware Required:

* Arduino board
* 5 x PNP transistor ([**2SA1015**](http://www.unisonic.com.tw/datasheet/2SA1015.pdf), [**2N3906**](https://www.onsemi.com/pub/Collateral/2N3906-D.PDF) …)
* 2 x 100nF ceramic capacitor
* 5 x 4.7k ohm resistor
* 2M ohm resisor
* 100k ohm resistor
* 10k ohm resistor
* 1k ohm resistor
* 100 ohm resistor
* 330 ohm resistor
* 10k ohm variable resistor or potentiometer
* Breadboard
* Jumper wires

Arduino autoranging ohmmeter circuit:  
Project circuit diagram is shown below.  
Note that all the grounded terminals should be connected together.

Diagram, schematic

Description automatically generated

The resistor which we want to measure is connected between Arduino analog channel 1 (A1) and GND as shown in the above circuit diagram (Ohmmeter Probes). The 100nF capacitor is used to stabilize the voltage across the unknown resistor.

The five PNP transistors Q1 ~ Q5 are general purpose transistors and they are of the same type 2SA1015, 2N3906 or equivalent. They are used as electronic switches for our autoranging ohmmeter.  
The emitter terminal of the five transistors are connected together to 5V pin of the Arduino board.

The collector of each transistor is connected to a different resistor. At any time there is only one ON transistor whereas the others are OFF. For better accuracy, each of the 5 resistors should have tolerance of 1% or lower.  
Also, each transistor base is connected to Arduino digital pin through 4.7k ohm resistor.

The AREF pin of the Arduino board should be connected to 5V pin with 100nF capacitor between it and GND pin.

# Master Controller

Since the master controller will require 2 separate UART connections, one for connecting to the ohm meter and another for connecting to the robot arm unit, using an ATMEGA328p based system would not be sufficient. Either an ESP32 (3 UART) or RP2040 (2 UART) based system would overcome this obstacle, both systems can be programmed using Arduino or Platform IO, as well as “bare metal” using their own respective SDKs.

# Electromagnet

A picture containing graphical user interface

Description automatically generated

<https://wiki.keyestudio.com/KS0320_Keyestudio_Electromagnet_Module_(Black_and_Environmental-friendly)>

* Working Voltage: DC 5V
* Working current: 0.3A (maximum)
* Maximum power: 3W

The above electromagnet can be controlled using a PWM signal on its SIG pin, the larger the PWM value the larger the magnetic field (up to 1kg, this needs to be tested).