Diamonds can picker

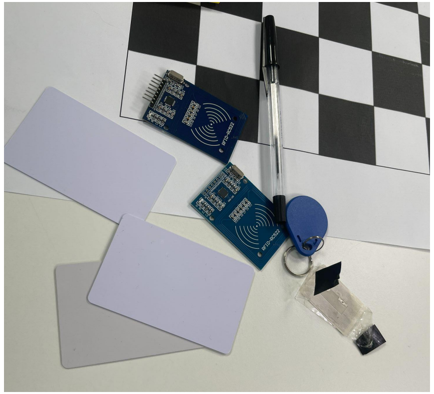
M00851040 Alin Ivan
M00826150 Johanna Amunyela
M00841284 Mohammed

Introduction

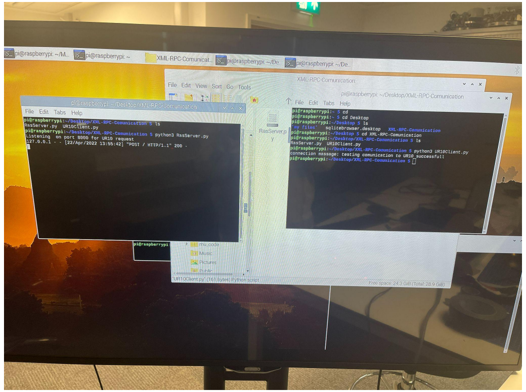
- Problem
  - Manual tin scanning and normal printed barcodes
  - Diamond security in the recovery Plant
- Objectives
  - Create an integrated robotic system using,
    - UR10 robot arm
    - Camera Vision
    - RFID readers and tags,
  - To develop a more robust system in comparison to the current manual barcoding process

RFID

- **Set up and components used**
- Raspberry pi 4 GPIO
- SPI protocol
- RC522 RFID Module
- Python Code
- Different RFID tags



- **Integration and additions**
- Soldered the RC522 Module
- Tested the wires using a Multimeter
- Measured the connectors and and wired the robot
- Using XML-RPC, I developed the request to read and write to the tag
- UR10 logic , Translation equation, XML-RPC server and client , troubleshooting



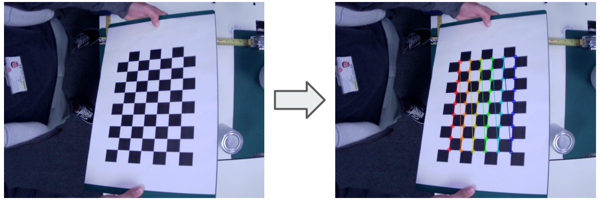
RFID

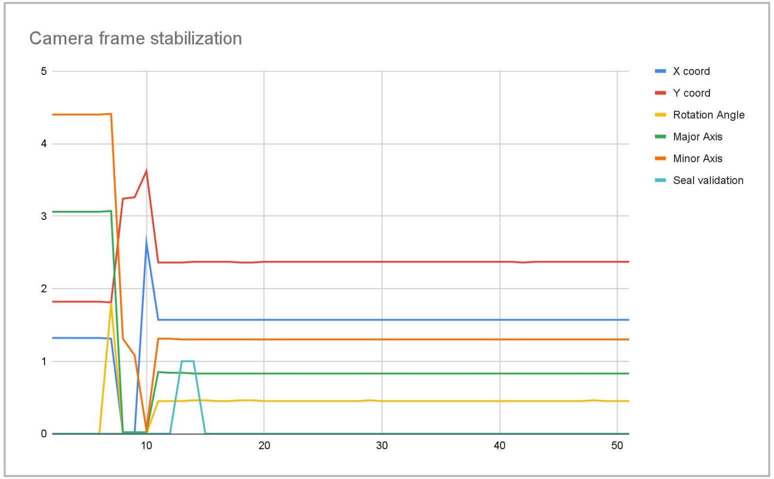
- **Challenges**
- RC522 only works on **13.56MHz** frequency tag and above
- Tin Metal Surfaces need special metal friendly rfid tag
- GPIO unstable due to the breadboard setup.
- Pulling Seven 5+ meter wires.

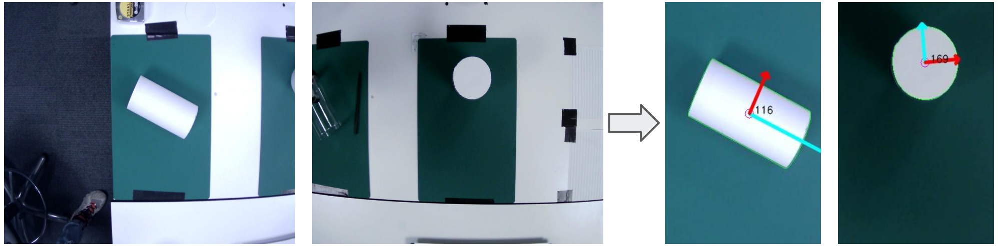
- **Additional components**
- Tin Tracking GUI
- Tin Tracking Database

Image processing

- Picking & caning cameras
- Python & OpenCV
- Camera calibration
- Frame stabilization
- Data extraction
  - Object contours
  - X, Y coordinates (mm)
  - Rotation angles in degrees
  - Major (MA) & minor (ma) axis (mm)
  - Seal validation (Boolean)





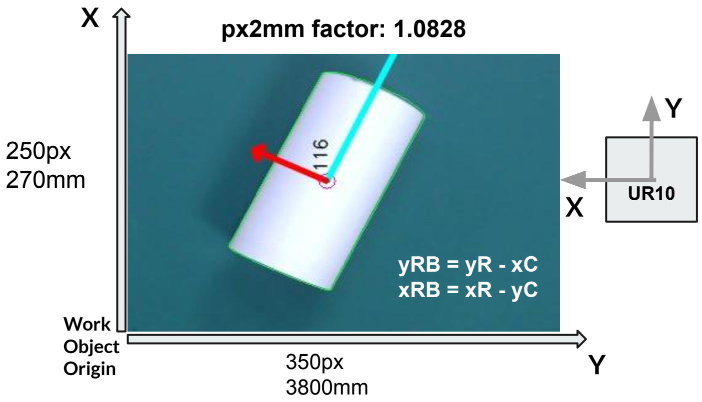



```
tin data: {'x': 106, 'y': 144, 'angle': 153, 'MA': 112, 'ma': 243}
seal: True
rfid: True
```

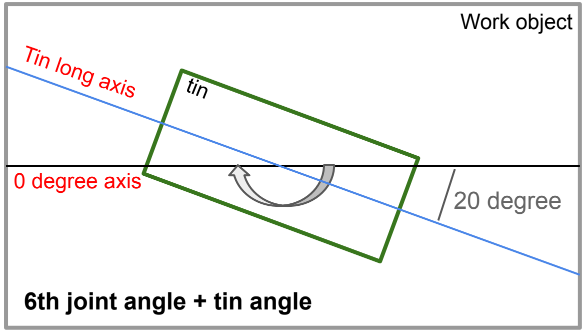
Computer vision integration

- **Work object reference**
  - Frame crop (px)
  - Physical measurement (mm)
  - Units of measurement conversion (px to mm factor)
- **Rotation angle (using joint angles)**
- **XMLRPC endpoints**
  - Next pose in activity sequence
  - Joint angles for wrist rotation
  - True/False for seal validation

Work object dimensions and x, y coordinates relation between the robot and the camera



Angle rotation measurement approach



Server endpoint to provide wrist rotation data in form of joint angles vector in respect of image processing results

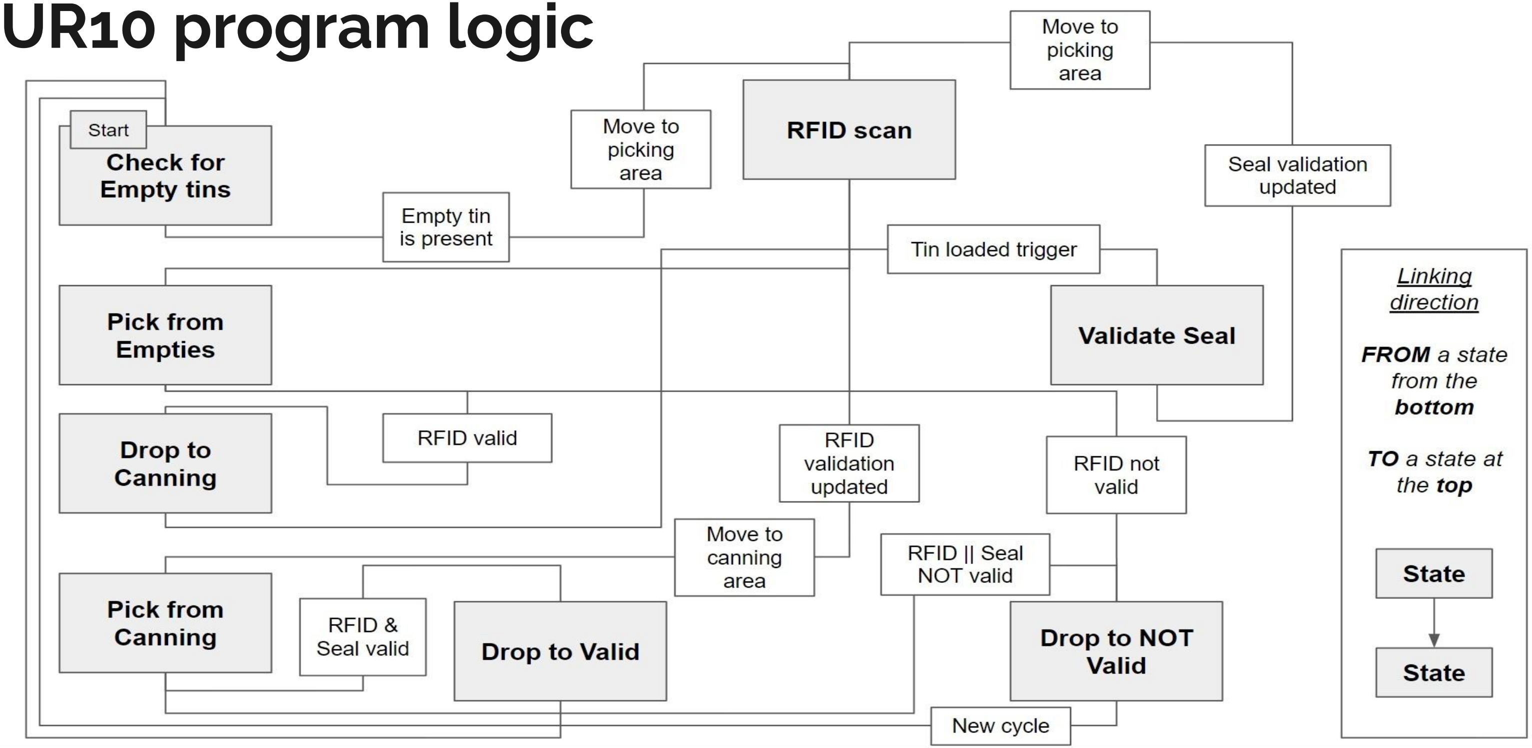
core.py

```
def requestGripperRotationJointAngles(q, a):
    return [q[0], q[1], q[2], q[3], q[4], d2r(tinAngle + r2d(q[0]) + a)]
```

main.py

```
server.register_function(requestGripperRotationJointAngles, "requestGripperRotationJointAngles")
```

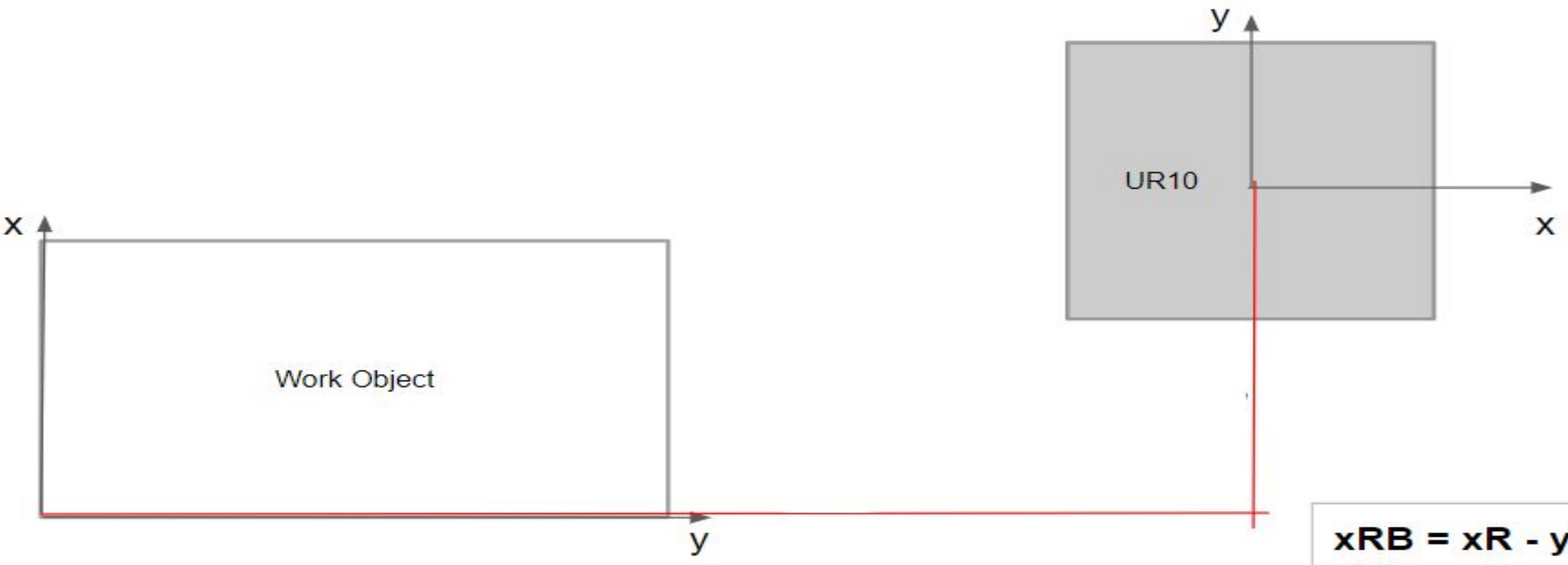
UR10 program logic



UR10 integration

- **Coordinate frames were established in relation to the work objects by:**
  - Relating to the X, Y coordinates of the picking work object to the robot, xRB = xR-yC and yRB=yR-xC (A more detailed explanation in the following slide)
  - Using free drive and pose data into Raspberry Pi variables
- **XMLRPC requests ensured pose data is being transferred to the robot:**
  - Achieved by calling the server endpoints
  - Requested data: Pose, joint angles (wrist rotation), true/false (seal & rfid validations)
  - Forward kinematic helped to transform from joint angles to pose
  - Endpoints requests: Get current tcp pose, get current joint angles
- **Actuation**
  - MoveJ, MoveL, RG6 URCap gripper opening and force

Relation between the origin of the Robot and Camera view



xRB = xR - yC
yRB = yR - xC

R - Robot
C - Camera

z = 0, and does not require transformation

Demonstration

TEST	RESULT
Picking: Scanner reach RFID tag from 1st attempt	Passed - rotated to 1st position
Picking: Scanner reach RFID tag from 2nd attempt (180°)	Passed - rotated 180° to 2nd position
Picking: Tag in close position in relation to scanner	Passed - scanned and tin picked
Picking: Tag in far position in relation to scanner	Passed - scanned and tin picked
Picking : Tag not present	Passed - tin picked and placed in packing not valid area
Caning: Seal valid + tin valid	Passed - tin picked and placed in packing valid area
Caning: Seal valid + tin not valid	Passed - tin picked and placed in packing not valid area
Caning: Seal not valid + tin valid	Passed - tin picked and placed in packing not valid area
Caning: Seal not valid + tin not valid	Passed - tin picked and placed in packing not valid area

Conclusion and further work

Conclusion

- Robotic system is successfully integrated to include vision, tin tracking and actuation
- Testing was mostly successful with issues concerning robot actuation and computer vision
- Consistency issues related to empty tin presence recognition
- Flexibility issues related to rfid scanner position

Further work

- Additional validation criteria can be added for the seal such as color spectrum processing
- Adding complexity in contour analysis should improve tin presence detection
- Using a proximity sensor for the rfid scan approach can improve can size operating range
- A scale in canning area could improve tracking if related to previous processes (diamonds weight)
- Better processing of z axis could facilitate processing various tin sizes in a tilted environment
- Using variable gripper data can improve the system versatility in terms of can size or composition