# Final Presentation

Ezra Bakatubia, Aima Ayaz

This project is about transforming the PiCar X into the Autonomous Library Assistant. The goal is to build a small robot (Pi Car) that can move on its own, find items using a camera, and interact with them using a gripper. This presentation will explain how we designed, built, and tested the system.

# What we do

Why we wanted to build this.

- Inventory checking is boring, slow, and repetitive
- Robots can handle these tasks accurately
- Helps workers focus on more important tasks

Inventory checking in libraries or storage areas takes a lot of time and can get repetitive. Workers often have to manually scan or check every item on every shelf. The A L A is designed to automate that boring part of the job. By letting a robot handle repetitive tasks, humans can focus on helping people, organizing materials, and doing tasks that require judgment and creativity.

# Hardware Design

1

To transform the PiCar X into the A L A, we added new hardware.

2

The main upgrade is a servo-powered gripper mounted on the front, which acts like a small robot hand.

3

We use the PiCar's camera for vision and the ultrasonic sensor to measure how close the robot is to objects. Everything is controlled by the Raspberry Pi, which runs our code.

# Technicality

## Hardware Systems

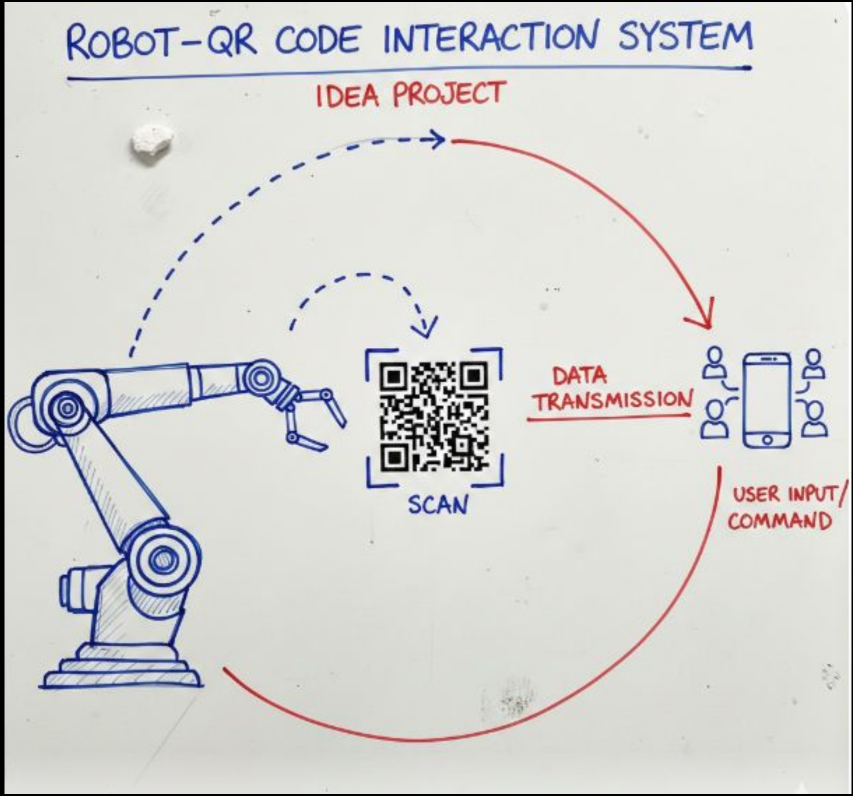
- **Base Robot:** PiCar X
  - Motors for movement (drive and steering)
  - Strong frame for stability
  - Raspberry Pi computer for onboard processing
  - Camera for visual input
- **Gripper System:**
  - Small servo-powered gripper attached to the front
  - Controlled via Raspberry Pi through Robot HAT
  - Functions: grab items, tap/check items
- **Sensors:**
  - **Camera:** main visual sensor for QR code detection
  - **Ultrasonic Sensor:** measures distance to objects for:
    - Stopping at the right spot
    - Collision avoidance
- **Power & Connectivity:**
  - All actuators and sensors powered through PiCar X's existing power setup
  - Raspberry Pi interfaces with all peripherals

## Software Systems

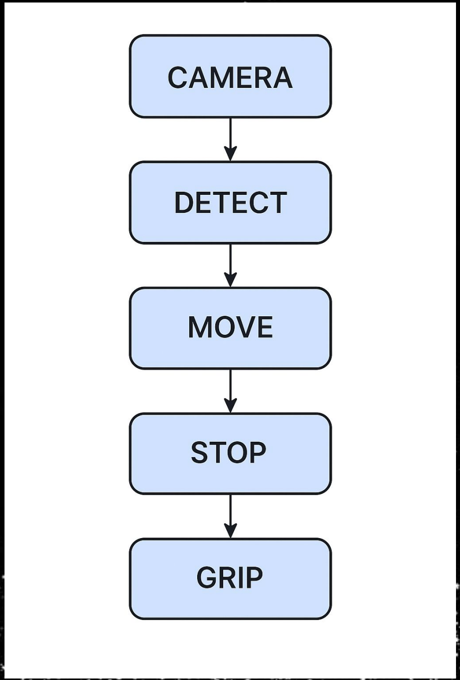
- **Programming Environment:**
  - Python as the main language
  - Libraries:
    - **OpenCV:** computer vision for object detection, frame processing
    - **ZBar:** QR code scanning and decoding
- **Control Systems:**
  - **PD (Proportional-Derivative) Controller:** smooths steering and navigation toward QR codes
  - **Distance Threshold Logic:** triggers gripper and QR code reading when within target distance
- **Navigation Phase:**
  - Continuous video frame capture
  - Detects QR code location relative to screen center
  - Calculates steering corrections using PD controller
  - Moves forward while keeping target centered
- **Interaction Phase:**
  - Triggered by ultrasonic sensor detecting close proximity (e.g., 5 cm)
  - Stops robot precisely at target
  - Reads QR code with ZBar
  - Logs item number to file (inventory record)
  - Activates gripper to interact with the item

# Problem & Idea

We wanted to solve the problem of slow and inefficient inventory checks. The idea is simple: make a robot that can drive down an aisle, spot the QR code on an item, stop in front of it, read it, and record that the item exists. This allows libraries to stay organized automatically without constantly sending people to scan everything.



# Software Pipeline




## Phase 1

The A L A works in two phases. First, the navigation phase, where the robot looks for a QR code and steers toward it.

Learn more

## Phase 2

Second, the interaction phase, where it stops at a safe distance, reads the code, and closes the gripper. These phases run in sequence to complete one full inventory check.

Learn more


## Computer Vision

The camera constantly collects video frames. Our program uses OpenCV to detect and track a QR code in the image. The robot calculates how far the QR code is from the center of the camera view. If it's off-center, the robot steers left or right until the QR code is lined up correctly.

```python
# Initialize camera and sensors
camera = Camera()
ultrasonic = UltrasonicSensor()
gripper = Gripper()
qr_decoder = ZBarDecoder()  # or any QR code library

# Safety parameters
EMERGENCY_STOP_DISTANCE = 8  # cm
TARGET_DISTANCE = 5  # cm

# Main loop
while True:

    # 1. Check for obstacles first
    distance = ultrasonic.read_distance()
    if distance < EMERGENCY_STOP_DISTANCE:
        stop_robot()
        continue  # skip scanning if something is too close

    # 2. Navigate towards target QR code (navigation phase)
    frame = camera.capture_frame()  # capture only current frame
```

```python
# 3. Detect QR code only
qr_code = qr_decoder.detect_qr(frame)  # returns None if no QR code
if qr_code:
    # Compute QR code position relative to frame center
    steering_offset = compute_offset(qr_code.center)
    adjust_steering(steering_offset)
    move_forward_safely()

    # 4. Interaction phase: check if robot is close enough
    if distance <= TARGET_DISTANCE:
        stop_robot()

        # Read and log QR code (only data, no image storage)
        item_id = qr_decoder.decode(qr_code)
        save_item_to_inventory(item_id)

        # Interact with item
        gripper.close()
        wait(1)  # hold gripper closed briefly
        gripper.open()

else:
    # No QR code found, keep scanning
    search_for_qr_pattern()
```

# We followed a clear development process. First, we came up with the idea. Then we designed the hardware and picked the sensors. After that, we coded the vision and control systems. Each part was tested separately to make sure it worked. Finally, we combined everything into one fully working robot and evaluated its performance.

Following the development pipeline.

1. Ideation
2. Design
3. Implementation (vision, control, planning)
4. Testing modules separately
5. Integration
6. Evaluation

# Evaluation

To see how well the A L A performed, you would need to measure for accuracy, response time, and stability.

For accuracy, we you would measure the distance between the gripper and the QR code after the robot stopped. A ruler would be placed on the floor, and we recorded the error. Our goal was less than one centimeter.

For response time, you time how long the robot took from detecting a QR code to fully stopping.

For stability, check if the robot stayed steady, avoided shaking, and gave repeatable results across multiple trials.

The A L A shows how robots can support people in everyday environments. It makes inventory work faster and more accurate while reducing stress on employees. Ethically, the robot must always be safe, slow, and aware of its surroundings. The camera must only be used to read QR codes, not to watch people. The project highlights responsible robotics: helping humans without replacing or risking them.

# Thank you

Thank you professor for a great semester!

## Ready for what's next?

Let's talk

Ezra Bakatubia
682-408-6965