

MAD76 Vision

Frank Tränkle*
Hochschule Heilbronn, Germany

September 20, 2025

*frank.traenkle@hs-heilbronn.de

Contents

1	Agenda	3
2	AruCo Markers	4
2.1	Marker Generation	5
2.2	Frame Markers	8
2.3	Car Markers	10
3	Camera	12
3.1	Mounting Camera	13
3.2	Focus and Aperture	15
3.3	Camera Calibration	18

1 Agenda

- Print and place the AruCo markers for computer vision (see Section 2)
- Mount, adjust and calibrate the Raspberry Pi camera (see Section 3)

2 AruCo Markers

MAD76 applies ArUco markers in computer vision for detecting and tracking cars. This section explains

- how to generate and print the markers (see Section 2.1),
- how to place the coordinate frame markers (see Section 2.2),
- how to place the car markers (see Section 2.3),

2.1 Marker Generation

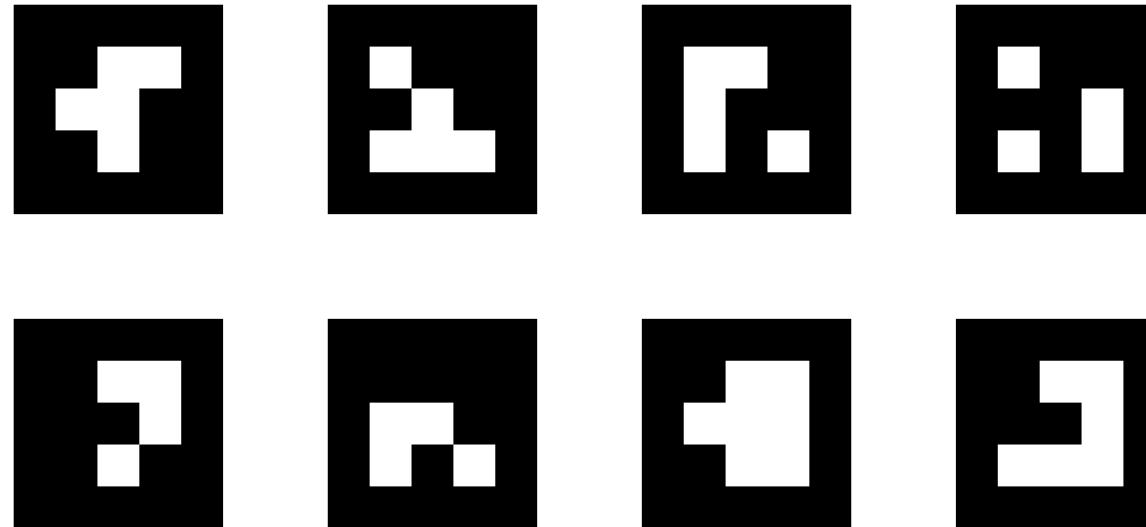


Figure 1: AruCo markers for cars and coordinate frame.

- The cars are tracked by ArUco markers [1].
- Computer vision computes the Cartesian coordinates and the yaw angles of the cars.
- The markers are generated with the OpenCV ArUco library.
- A custom ArUco dictionary of 8 markers with a size of 3x3 bits is used to increase the reliability of computer vision.
- The PNG image of the 8 markers can be optionally created by

```
cd ~/src/mad2/mad_ws  
install/mbmadvisionaruco/lib/mbmadvisionaruco/create_board --bb=1 -d=17 -w=4 -h=2 -l=200 -s=100 markers.png
```

- The markers IDs are from 0 to 7, 0 to 3 in the first row from left to right, and 4 to 7 in the second row.
- Print the markers in Fig. 1 on a snow-white, 80 grams paper.
 - Make sure to configure high quality printing.
 - Scale the printing such that the black area of the markers have a height and width of 21mm each.
- Cut the markers as squares including approx. 5mm boundaries.
- Note the marker IDs before cutting with a thin pencil on the boundaries, because you will need these IDs later on.

2.2 Frame Markers

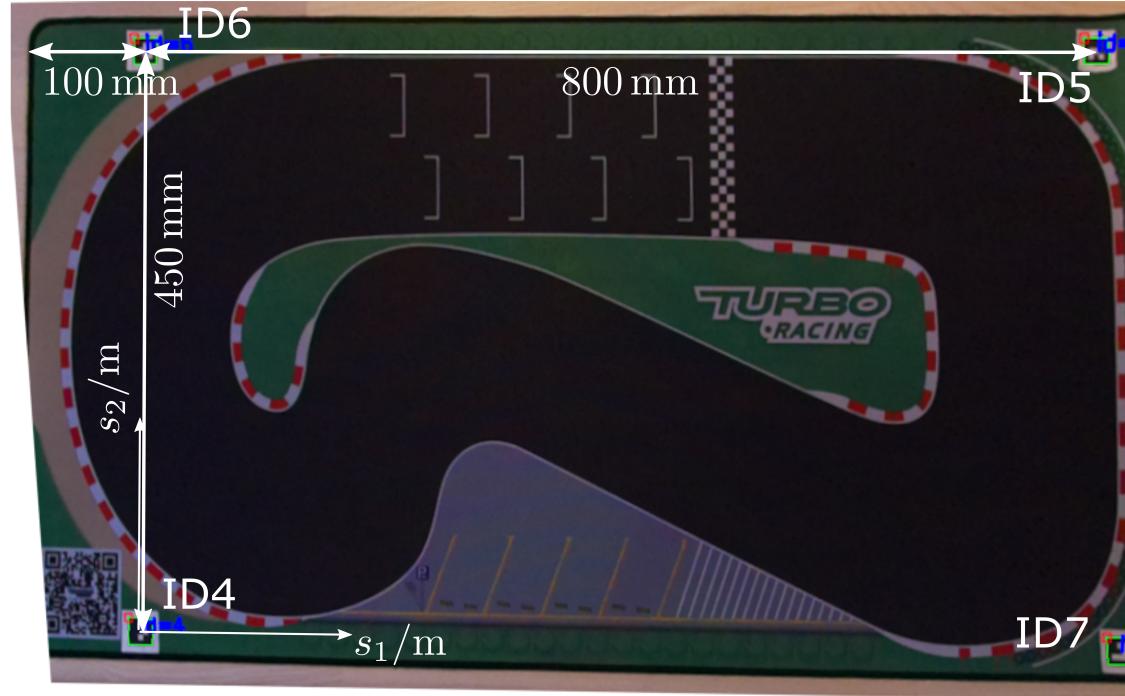


Figure 2: Track with 4 coordinate frame markers.

- 4 frame markers define the coordinate frame of the track.
- All coordinates of cars and track are measured in meters.
- The frame origin $(s_{01}, s_{02}) = (0\text{m}, 0\text{m})$ is at the center point of marker ID4.
- Place frame markers with IDs 4, 5, 6, 7 at corners of board as depicted in figure.
 - It is recommended to place the markers with high accuracy in the 1mm range. Otherwise, the control functions of the MAD76 driving stack will lose precision.
 - Although modified distances may be later configured in the ROS2 package `mbmadvisionaruco`.
 - The distances are measured at the marker center points.
 - The markers must form a rectangle.
 - The sequence of the marker IDs is essential.

2.3 Car Markers



Figure 3: Red car with marker ID 0.

- Each car has its individual marker.

- The following configuration is recommended:
- | Marker ID | Car |
|-----------|--------------|
| 0 | red orange |
| 1 | green yellow |
| 2 | blue |
| 3 | white |

- If you have fewer than four cars, please start with ID 0 in any case.
- Each marker's center point must be placed exactly at the car's rear axle center point.
- The horizontal orientation of the marker must match to the forward direction of the car.

3 Camera

This section explains

- how to mount the camera (see Section 3.1),
- how adjust focus and aperture (see Section 3.2),
- and how to calibrate the Raspberry Pi camera using ChArUco boards (see Section 3.3).

3.1 Mounting Camera

- The camera must be mounted above the MAD76 track for bird's eye view.
- The camera lens should be at a height of approx. 106cm above the track.
- The ideal position is above the center of the track.
- If the camera is mounted on a tripod next to the track, a tilt in one direction is necessary, which should be as small as possible.
- In all other directions, the camera should be aligned in parallel to the track and not be tilted.

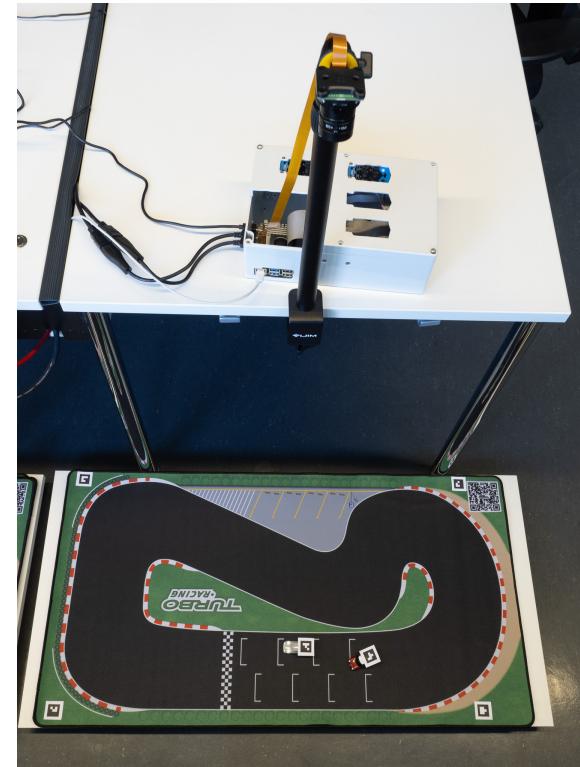


Figure 4: Mounting camera above the MAD76 track.

- Place the track as close as possible to the tripod, such that
 - the camera can capture the entire track without any obstruction,
 - the whole track area is as close as possible to the camera for higher pixel resolution,
 - and is in the focal plane of the camera
- For adjusting the position and tilt of the camera, run the following ROS command

```
ros2 launch mbmad madpiman.launch
```

This opens a window with the camera image as depicted in Figure 2

- All 4 frame markers and track borders shall be visible.
- The lower and upper track borders shall be in parallel to the image borders.
- The lower track border shall be as close as possible to the lower image border if the camera is titled.

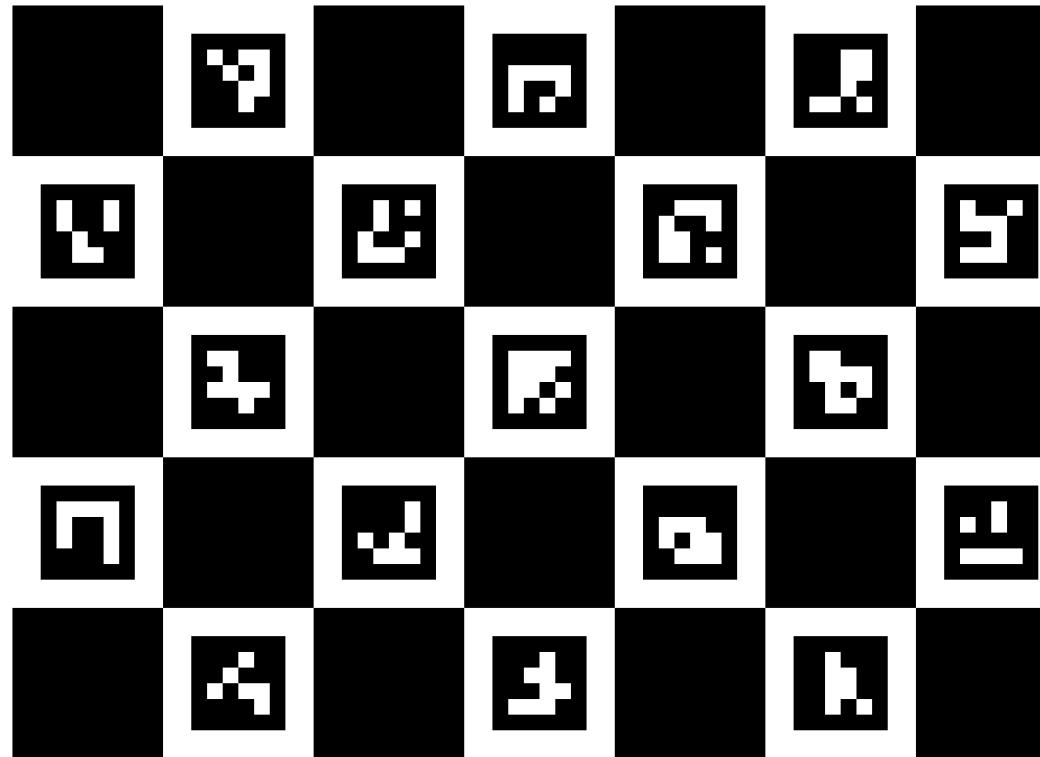
3.2 Focus and Aperture

- In MAD76, the camera acquires images with a frame rate of 40 frames per second (fps) (or a sampling time of 25ms)
- The MAD76 SW sets the exposure time to a fixed value of $2\text{ms} = 1/500\text{s}$ to avoid motion blur.
- The focus of the lens must be adjusted to ensure that the entire track area is in sharp focus.
 1. Fasten the screw for aperture adjustment (OPEN<>CLOSE).
 2. Loosen the screw for focus adjustment (NEAR<>FAR).
 3. Adjust the focus by turning the focus ring until the entire track area is in sharp focus.
 - You may zoom in by hitting the menu button + of the camera image window.
 4. Fasten both screws.
- The aperture must be adjusted to the small value (large f-stop number) to achieve a large depth of field.
 1. Fasten the screw for focus adjustment.
 2. Loosen the screw for aperture adjustment.
 3. Adjust the aperture by turning the aperture ring, such that all 4 frame markers are reliably detected.
 - The detection is successful if all 4 frame markers are highlighted in the camera image by green bounding

boxes.

- A darker image is better than a brighter image for reliable detection and greater depth of field.
4. Fasten both screws.

3.3 Camera Calibration



The Raspberry Pi camera must be calibrated, so that the MAD76 can undistort the camera image frames [2]. The calibration is performed applying an ChArUco board, which is an augmentation of a chess board by Aruco markers for higher precision. Follow the following steps for calibrating your camera:

- Print the marker board in Fig. 5 on a snow-white DIN-A4 paper. Use high-quality printer settings.
- This PNG image can optionally be created by

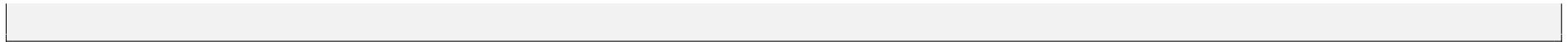
```
cd ~/src/mad2/mad_ws  
install/mbmadvisionaruco/lib/mbmadvisionaruco/create_board_charuco -d=0 -w=7 -h=5 -ml=500 -sl=800 charucoboard.  
png
```

- Fix this paper on a cardboard.
- Measure the side lengths of the squares and the Aruco markers (not including the white boundaries) in meters.
- Calibrate the camera by running the following command:

```
ros2 run camera_calibration cameracalibrator --pattern=charuco --size 7x5 --square 0.036 --charuco_marker_size  
0.022 --aruco_dict 4x4_50 image:=~/mad/camera/image_raw camera:=~/mad/camera camera/set_camera_info:=~/mad/  
camera/set_camera_info
```

- After successful calibration the camera matrix and distortion coefficients are stored in the file

```
~/.ros/camera_info/imx296__base_axi_pcie_120000_rp1_i2c_88000_imx296_1a_800x600.yaml
```



or similar.

- This calibration data file will then be automatically loaded by the MAD76 computer vision for undistorting camera frames.

References

- [1] OpenCV. *Detection of ArUco Markers*. Accessed: 2025-02-05. 2025. URL: https://docs.opencv.org/4.x/d5/dae/tutorial_aruco_detection.html.
- [2] ROS. *How to Calibrate a Monocular Camera*. Accessed: 2025-02-05. 2025. URL: https://wiki.ros.org/camera_calibration/Tutorials/MonocularCalibration.