

14 Week Development Timeline

Designed for 14 sessions, 1-2 hours per session.

Week 1. Prepare all software development environments.

- Install **Python 2.7** and all of its dependencies.
- Install **Arduino** and all of its dependencies.

Week 2. Gain an understanding of what is possible.

- Explore **example capabilities** of the Mark II robot.
- Research other prior art in the field of robotics.

Week 3. Define a task you want the Mark II robot to do.

- Think about how humans solve the task.
- Draw sketches of grippers/attachments.

Week 4. Learn the fundamentals of Python.

- **Learn the basics of Python.**

Week 5. Go over basic setup and operation of the robot.

- Learn about the robot's **features** and limitations.
- Maintenance and common issues.

Week 6. Get familiar with writing code for the robot.

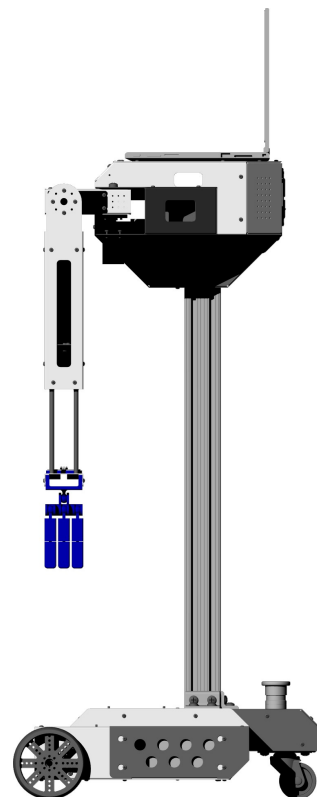
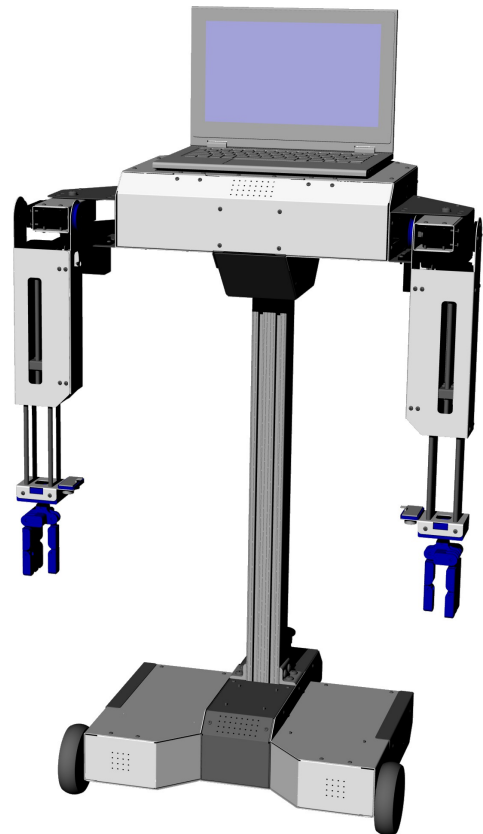
- Write code for a simple "dance" routine.
- Observe how the robot responds to changes in code.

Week 7. Computer Aided Design for grippers #1.

- Design attachments according to drawings.
- Each gripper can have up to 5 servomotors.

Week 8. Physically produce the grippers #1.

- 3D Print grippers out of plastic.
- Use Screws, Legos and other parts to put it together.



Week 9. Programming the robot to use the grippers #1.

- Code robot motions to achieve the task.
- Work on gripper servo articulation.

Week 10. Computer Aided Design for grippers #2.

- Analyze how the 1st set of grippers can be improved.
- Make improvements to the grippers according to analysis.

Week 11. Physically produce the grippers #2.

- 3D Print grippers out of plastic.
- Use Screws, Legos and other parts to put it together.

Week 12. Programming the robot to use the grippers #2.

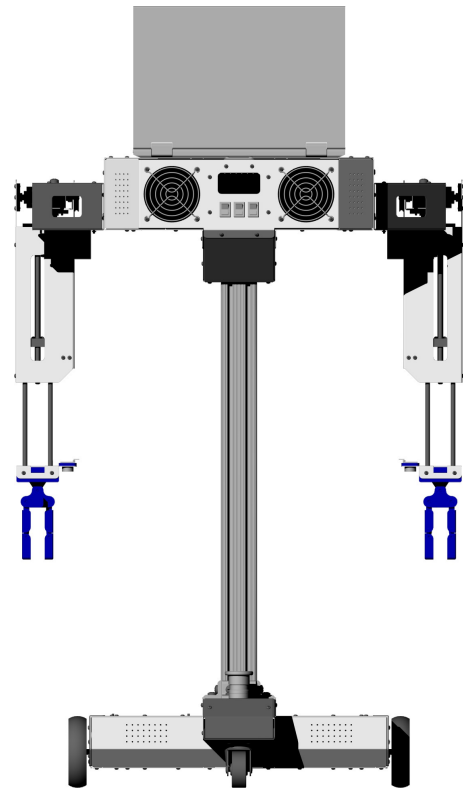
- Improve robot code to achieve the task.
- Fine-tune motor control and articulation.

Week 13. Compile and edit documentation for publishing.

- Aggregate and organize photo and video documentation.
- Edit and render a clean video showing your work.

Week 14. Write a final report and publish documentation.

- Describe what worked and what didn't.
- Give **feedback** on how to improve the program the future.



Note that this timeline represents guidelines as opposed to rigid curricula, and as such is very flexible to changes. In addition, this timeline is only an estimate, and actual times may vary greatly depending on learning and development speed.

