# Bishop’s Knights at Home

We do not yet have a functioning robot. We have been working on developing components that will be needed to build a successful robot. We will have more time moving forward, most of the team has been taking AP exams, but the last AP exam for them is May 25th.

Here is a list of components that are currently working, separately:

- **Construction:** We are moving to a custom-built robot. We have the following components:
  - *Controller:* Nvidia Jetson Xavier NX running JetPack 4.5, which is based on Ubuntu 18.04 LTS. This controller includes a GPU, dual Raspberry Pi Camera ports, RPi GPIO pinout, improved ARM CPU (compared to the RPi and Jetson Nano), WiFi, Ethernet and Bluetooth. To improve speed, we have installed and moved the filesystem to an NVMe drive in the M.2 Key M slot. For software installed (OpenCV, Tensorflow, Pytorch and much more), see this Github repository. We have also been able to run the Pose Detection tensorflow algorithm on a Google Coral USB accelerator in case we need to free up the GPU and RAM for other tasks.
  - *Intel RealSense cameras:* We have a D435i Stereoscopic depth camera and a T265 Tracking Camera. We have libraries for these cameras installed, tested and working (librealsense 2.41.0, plus pyrealsense2 for Python 3.6).
  - *Cameras:* We have several PiCameras (ver. 2) for computer vision, and know how to use them on the Jetson.
  - *Sensors:* We have IR range-finding sensors, both VL6180 and VL53L0X (from Adafruit) and know how to use them on the Jetson. We have a variety of other sensors if any are needed.
  - *Motors:* We have several Dynamixel XM430-W210-T, XM430-W350-T, and XL430-W250-T for the base and the arm, and U2D2 for USB control, and U2D2 power hub to supply power. We have installed and used the Dynamixel SDK on the Jetson to control the motors.
  - *Battery/Power:* We have 3-cell Lithium-Polymer batteries (11.1-12.4V, surge power up to 20A) that can supply power to all of the motors, and a step-up voltage regulator to get the 19V for the Jetson Xavier NX controller.
- **Pose recognition:** Uses computer vision to recognize the location of limbs of a person in view of the camera. Specifically, it can test to see if a person is raising their hand to get the robot’s attention. This uses a custom-built system, using Tensorflow, made by one of the students on the team. The student developing this is working on tracking the centroid of the person’s body and using this to improve continuity of the limb tracking from frame to frame. As mentioned earlier, we can push the computation out to the Google Coral USB accelerator in order to free up resources for other tasks if needed; on the Coral, this runs at 640x480 at ~30fps with very low latency.
- **SLAM:** We have implemented Intel’s occupancy-mapping algorithm (c.f. intelrealsense Github) using the Realsense cameras/IMU, which we have working even though it was designed for a much earlier version of librealsense. We are running that through ROS

Melodic; we are comfortable setting up a subscriber to fetch the map. We have not yet implemented a path-finding algorithm.

- **Robot Design:** We have a framework for the robot design, with several parts 3D printed. We have plenty of very sturdy 5mm plexiglass and a good laser cutter for making the robot. We have plenty of hardware: nuts, bolts, standoffs, support rods, etc..
- **Text-to-speech and speech-to-text:** Last year, we implemented Sphinx for speech-to-text, and had that working well. We also had speech-to-text working well, we plan to re-implement these essentially as they were.

We expect our first task to be ‘Fetch my luggage.’ We have about half of the individual tasks already completed: pose analysis, speech-to-text, text-to-speech for the interaction with the person requesting help, mapping to find the luggage. We will need some more computer vision to identify the object as luggage and a path-finding algorithm. We will need to finish developing the robotic arm to grab the luggage, and set up a system to allow the robot to follow the person.