Distributed Climate Control - Interface Specifications
By Dan Coen and Conor Sheridan

OpenCV Dependency: Many modules of this project depend on OpenCV. An install of version 3.0 or higher will be sufficient for running our code. To install OpenCV, visit the following link: https://docs.opencv.org/master/d9/df8/tutorial_root.html. Please note that the Pylepton library also makes use of the python wrapper for opencv, which should be installed as well.

Flir Lepton

Python

Class Socket: Contains a simple fixed length server for the raspberry pi to send thermal images to the client. Uses Python INET sockets.

server_init(self, sock=None) - Create new socket, unless given existing socket

raspi_connect(self, host, port) - Connect to given socket with this self socket

raspi_send(self, msg) - Send image data to client. Continues to send data until length of msg is sent. Raises runtime error if connection is broken during sending of message.

raspi_receive(self) - Receive message from client. This will be used to get frame requests or desired I2C commands from the client. Receive messages with frame size of 2048. Returns bytestring of image. Raises runtime error if attempt recv returns length 0 string.

Pylepton: Groupgets Impementation of simple lepton library.

capture(flip_v, device) - Takes a picture using pylepton library, returning an normalized 8 bit image in unsigned integer format. Set Flip_v = True to flip the image horizontally, and device = “/dev/spidev0.0” if using spi port 0 (adjust accordingly). Should be checked for acceptable data.

Kinect

C++

These are the main classes involved that we will be using for depth, motion, and body tracking, in that order:

Depthframe Class - Represents a frame where each pixel represents the distance (in millimeters) of the closest object seen by that pixel.
CopyFrameDataToArray - Copies the depth frame data into the array provided.

Bodyframe Class - Represents a frame that contains all the computed real-time tracking information about people that are in view of the sensor.
GetAndRefreshBodyData - Gets refreshed body data.

BodyIndexFrame Class - Represents a frame that indicates which depth or infrared pixels belong to tracked people and which do not.
CopyFrameDataToArray - Copies the body index frame data into the array provided.

On all of these classes, the close() method will also be used, which simply releases system resources associated with the frame.

To initialize each reader stream and the sensor itself, you would do it as such, with the body reader as an example:

```
KinectSensor^ sensor = KinectSensor::GetDefault();
   sensor->Open();
   bodyReader = sensor->BodyFrameSource->OpenReader();
```

Python

PyKinect2

PyKinect2 allows us to write Kinect applications in Python, which we are using to have a more unified and simplified project. The requirements are pretty straightforward as listed with the link above, and requires the Kinect for Windows SDK that we are already using.

We are running the Kinect for Windows SDK v2 on our UP board, which is running Windows 8.1 We are getting all our Kinect data directly to this board and processing it with opencv, as described at the top. The Lepton data is also being sent to this machine as well.

Arduino Commands

C++ - Serial Communication (Windows)

Serial Class: An object used to interface with the arduino by connecting via the coms port on windows.

ardSetPosition(int position) - Function to set the position of the pan and tilt bracket. The position corresponds to the portion of the room to be viewed. Returns 1 on error.

```
Ex. |0|1|2|3|
    |4|5|6|7|
```

C++ - Serial Communication (Arduino)

setPosition(int position) - Callback function for serial read when checking for command from windows machine. Sets pan servo and rotate servo appropriately for position given. Returns 1 on error.