Vision
Vision Setup

- The USB cameras in the Botball kit will work in either of the CBC’s USB ports
- Plug in a camera and you will be able to see the camera image by going to the vision screen
  - If you unplug the camera, the CBC may no longer recognize it if you plug it back in
    - You will need to restart the CBC if this happens
HSV Color Selection Plane

- Hue=0
  - Sat=0
  - Val=224
- Hue=360
  - Sat=0
  - Val=224
- Hue=0
  - Sat=224
  - Val=224
- Hue=360
  - Sat=224
  - Val=0

Note: 224 is the range of values the camera pixels put out in each of R, G & B.
Color Blobs

- Each pixel on the screen has an HSV color
- When we say red, we really mean a range of HSV colors on the color selection plane that are approximately red
- A rectangular piece of the color selection plane that corresponds to being red specifies the range of HSV colors to be viewed as red by the CBC
  - This is called an HSV color model
- A red blob is all contiguous pixels matching one of the HSV colors in the red range
- A blob has a size, position, number of pixels, major and minor axis, etc.
- If you want to identify Botguy with the camera, you look for a big red blob
Vision System Color Models

- The CBC can handle 4 HSV color models simultaneously
- It can track a number of blobs from each color model
- It can display the video in any one of three ways
  - **Raw** (live video)
  - **Match** (pixels matching the color model are highlighted)
  - **Tracked** (highlights matching pixels and shows blob boundaries and centroids)
Color Vision Interface

Vision..Tracking Screen

- **Raw** image is displayed
- Color Model 0 is being manipulated
- The **Bottom Right** corner of the color selection box is being adjusted
- It can be moved **Left**, **Right**, **Up** or **Down**
Color Vision Interface

Vision Tracking Screen

- Matched image is being displayed
- Pixels that correspond to selected color region are shown highlighted
Color Vision Interface

Vision..Tracking Screen

- **Tracked** image is being displayed
- The bounding boxes of the tracked blobs are displayed, along with the centroid of each blob
Training a Color Channel

• Any color channel (in Match or Tracked mode) can be trained for tracking color blobs that match a given HSV color model by using the Vision..Tracking screen

• The default settings for color model 0 are for pixels that are approximately red, yellow for model 1, green for model 2, and blue for model 3

• Once set, the vision settings from training are retained
  – Default settings can be restored from the CBC Settings screen
  – Hint: if you are using the camera, shrink the selection box as small as possible for any color channels you are not using (reduces processing load)
Vision System Library Functions

\texttt{track\_update, track\_count, track\_x}

- The CBC library function \texttt{track\_update()} is a command that causes the CBC to capture the most recent camera frame for analysis
  - Frame analysis determines blob properties such as the \((x,y)\) coordinates of the centroid of the blob
- \texttt{track\_count(3)} provides how many (blue) blobs are being seen for the model 3 track
  - If the count is 0 there are no (blue) blobs
  - Blobs are numbered 0,1,2, \ldots from largest to smallest
- \texttt{track\_x(3,0)} for track 3, blob 0, returns the value of the \(x\) coordinate of the centroid of the largest (blue) blob
Image Coordinates

- The camera’s processed field of view is treated as an x-y (column,row) coordinate array
  - The upper left corner has coordinates (0,0)
  - The lower right corner has coordinates (159,119)
  - The CBC display does not show the camera’s full field of view
Selecting the Action to Perform

if-else

- For while, an action is performed so long as the condition check is true
- In contrast, for if-else, one action is performed if the condition is true and another if it is false
- Example:

  ```c
  if (track_count(0) > 0)
      { printf("There's a red blob\n"); }
  else
      { printf("Don't see a red blob\n"); }
  ```
- The if control structure is a special case of if-else
Example Using Vision Functions

// Train the camera so that it recognizes a red colored
// object for color channel 0
int main() {
    int x, y, color=0;  // set up for color channel 0 (red)
    while (black_button() == 0) //run till button is pressed
    {
        track_update(); // process the most recent image
        if (track_count(color) > 0)
        {
            // get x, y for the biggest blob the channel sees
            x = track_x(color,0); y = track_y(color,0);
            printf("Biggest blob at (%d,%d)\n",x,y);
        }
        else
        {
            printf("No color match in Frame\n");
        }
        sleep(0.2);  // give print time to register
    }
    printf("Program is done.\n");
}