Should the motor be spinning? This is checking for a stop motor flag MotorPWMSpeed=0 updateMotorSpeed=1 *turn LEDs OFF Continue Should motor speed be updated? This is checking for a speed change flag Yes Update motor PWM duty cycle, continue Continue Should the display be updated? (slit mask only) Send new numbers to the setClockDisplay function, continue Continue Is the slit mask being used? Index point registered? Index point registered? Yes Wait Continue Continue Motor spin-up mode? Motor spin-up mode? When the motor first spins up it has to go through a specific When the motor first spins up it has to go through a specific sequence to ensure the motor is started and stable sequence to ensure the motor is started and stable Yes No No Yes Run spin-up algorithm, Run spin-up algorithm, Continue Continue Rev motor for display algorithm? On rare occasions the index point is missed and caught on the second time around. When this happens the algorithm will get stuck repeating the same error. This gives a display slice time of 2X too long. I can check for this by looking for a larger than normal display slice time. If this condition is registered, the motor speed can be momentarily changed which fixes the display glitch Rev motor for display algorithm? On rare occasions the index point is missed and caught on the second time around. When this happens the algorithm will get stuck repeating the same error. This gives a display slice time of 2X too long. I can check for this by looking for a larger than normal display slice time. If this condition is registered, the motor speed can be momentarily changed which fixes the display glitch Yes Run rev motor algorithm, Run rev motor algorithm, continue Continue continue Continue Show beginning animation? Show beginning animation? Yes Yes Time to update display pixels? Which clock mode? animation algorithm, Beginning animation algorithm, continue *turn off LEDs *reset display pixel array Increment time Increment time variables based *nipkow clock values variables based on on user input, function time, continue continue *Calculate for next rotations Continue Continue Which display mode? Which display mode? Run slit mask spinning, while turning the display algorithm for current display OFF, rotation, continue Have display pixels been updated? Keeps motor spinning, while Index point registered? turning the LEDs OFF, continue Continue Call nipkow display algorithm, continue Check for user input Slit Mask Display Algorithm Fill fullFrameArray buffer for a full rotation void drawClockFrame(void) Read the previous rotation time from TIM3->CNT Reset the array prior to filling it resetFullFrameArray() Reset TIM3 → CNT to 0 Calculate display slice time (previous rotation time) / (total # of display slices) There are 60 display slices Set values for each of the the nipkow displays, display pixels index for each aperture, and colors for each of the 12 pixels FOR A clock hand takes up 1 display slice all 120 display slices Draw each display slice at the correct time in the rotation until all Calculate which of the 24 nipkow displays has aperture 0, the rest of the apertures are offset by 2 displays slices have been displayed (1 full rotation) X120 Calculate which pixel aperture 0 is over and assign its Set the entire LED display array the color of the color, the rest of the 11 apertures' pixels can be current slice, continue calculated based on their physical relationship to setDisplayAllOneColor(x, x, x) aperture 0. Each of the 12 apertures cover 1 full pixel ring. Each pixel ring has 120 pixels. If (clockFace) Each pixel occupies one display slice. The display has a resolution of 120x12 pixels. Should clock index LEDs be turned on? 12, 3, 6, 9 Nipkow display algorithm Void drawFullDisplayFrameNipkow12(uint16_t pieSlice, uint8_t currentSpot) PieSlice = time to display each pixel Set index LEDs in LED display Continue array, continue CurrentSpot = (for future features) X60 If (edgeRunner) LED animation that circles the outside of the display once every second Draw each display slice (current 12 pixels) at the correct time in the rotation until all slices have been displayed (1 full rotation). 'slices' don't work quite the same here as they do with the slit mask. There are still 120 stages/slices/rays/etc in the rotation, but to display pixels with the nipkow disk more than one 'slice' has to be considered. There are 12 apertures (holes) in the light mask. Each of these holes will fall in front of one of 24 individual 'zones' or Set edgeRunner LED in display 'displays' which comprise specific LEDs on the LED board behind Continue array, continue the spinning mask. The display is set the color of the pixel currently being displayed. This means that 12 pixels are displayed at different points around the display at any given time. Update display with LED display array There are 12 apertures on the nipkow disk. Each aperture covers one pixel ring There are 12 pixel rings on this display There is are 120 pixels on each pixel ring Wait until its time to display the next slice The displays resolution is 120x12 There are 24 'displays' behind the spinning mask Each aperture is spaced precisely 2 displays apart Only 12 displays are used at any given time, the others are set to off Continue to next slice Set the color of each of the 24 displays behind the spinning mask for the current display slice Slit Mask At the beginning of this for loop it is assumed we just crossed the index point X120 This algorithm reads from the fullFrameArray array to determine display colors for every pixel, display, color, etc for each display slice in the rotation X24 Read from fullFromArray and set the colors of the LEDs in the LED array for the current display Continue to next display Wait until its time to display the next slice Continue to next slice Nipkow Mask (index is on bottom) LED Board W/ Displays

0

0

0

0

0

0

Spinning Vision Algorithm Flow Chart