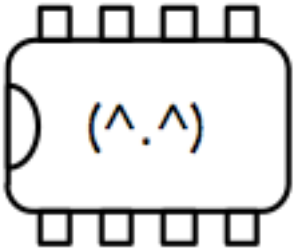
ZenAI Community of Robots ch4



www.ZENMCU.com

Draft 1 2023-02-24

zenai-0.0a #3

I spent some time reviewing my LED light sensing code with a view towards understanding why two identical LEDs seem to have widely varying sensing sensitivities. I made several observations.

First it turns out that when optimized with -Os my GPIO port code was getting optimized such that reading a pin returned the value of the entire port, not the pin.

I recoded it to read the port, mask the pin, and shift it to bit0, and that seemed to make it happy.

Second, there were several cases of write a 0 to a pin, write a 1, write a 0, etc to flip pins. But the compiler optimized the "dead stores" away and only kept the last one.

So I had to add the options -fno-dse and -fno-tree-dse which seems to have resolved the dead store problem.

I also noticed that my pin sampling and timing code was sampling the timer for the first LED and leaving the timer running, then sampling it again for the second led. So the second LED always showed a longer time than the first. So I sampled the timer once, then checked the LEDs, and this helped.

I also changed mu GPIO access from configuring the pins one after the other in sequence and instead configured them all simultaneously using the DIRSET, DIRCLR, and similar registers. That also helped make their timing more similar.

All of these things combined resulted in a fairly decent dual red LED led sensor that can tell if an emitting LED a few inches away is to the left, right, or centered. So I resurrected the vibrobot code and applied these changes. I also updated the servobot code.

[x] test the updated vibrobot code
    02/27/23 The vibrobot was already constructed from previous experiments. However it has blue LEDs so I did the new tests with a matching blue emitter. The result was that the robot was able to turn to face the emitting LED reasonably well. It wasn't perfect but it was a large improvement.

[x] test the updated servobot code
    02/27/23 This also showed improvement, though less than I expected. Part of the problem is that the servos want more than 3V, and they need to be pulsed enough times to accelerate them up to speed which tended to cause overshoot. Because my servo driver is intended to accelerate/decelerate smoothly between two 'positions' I think that two LEDs on a single servo for a head could work well. But the continuous rotation servos I used for the robot don't work that way; they are either on or off forward or backward for some duration.

I then applied the same types of changes to my bespoke RGB LED color sensor test, which showed the same sorts of improvements. It can now sense a similar LED and discriminate red vs green vs blue. However, it "sees" red the best, blue next, and green not very well (only very close and pointed straight on).
So I think that possibly at this point I may have enough of this worked out to make some small robots with red LEDs locate and approach each other. If I can get that working then I can add the ability for them to communicate.

Will I try to use the LEDs as the comm link? Probably, since LEDs are cheap and these

robots don't need to send large messages.

Will I try to use the infrared UART for this? Probably, since I know it works, and less code is required. But the IrDA demodulators are "expensive" (on the order of $5 in onesies) so if I can avoid using them I can make these toys cheaply.

End