

CS12020 Assignment 2 write-up – Shankly Richard Cragg – src27

I started by defining all the ASH values of different ASCII characters. I started by translating ASCII to ASH code so I created a function called `string2ashstring`. In this function I checked the ASCII (which I originally passed into the function from setup) characters against the values I declared using a large switch statement, and stored the ASH value in an array. Between each ASH letter I insert a `"/"`, but not at the start or end of an ASH word. The only tough part of this bit of code was ensuring the `"/"` didn't attach to the start or end of an ASH word, I got past this by adding an if statement so that nothing happened if the next character was a "space" or a "null" (Which represents the end of the string), and if this wasn't the case then to add the `"/"`.

With this done I started work on translating ASH input to ASCII, this was a lot tougher as depending on if the next character was a "space", `"/"` or a "null" a different action needed to be taken, and another action for if the character was not any of these things. Keeping track of 3 different buffers at the same time also proved a challenge, but by doing a trial run on paper of a short message it became easy to ensure they worked in sync. Once the logic for all the scenarios was taken care of through trial and error, I had to actually do the translation. I accomplished this by creating a new function called `ash2char`, which would use a string method called `strcmp` to see if the ASH letter was same value as the of the ASH value of different ASCII characters from earlier. If 2 came up a match it returned that letter. By putting all these into an array and then printing it I could now translate ASH to ASCII.

Next I worked on getting user input from the serial port. Using a function called `promptReadln` I put all the user input into an array with each character in a separate position, this allows me to look at the first character the user entered and determine if it was ASH or ASCII, and pass it to the correct function to do the translation. This was done by looking at what the first character in the array that the user input, and if it was an ASH symbol of `!"`, `""`, or `@`, then I assumed the whole thing was ASH, and if not then I assumed it was ASCII. Once this is done it then prints out whatever the user entered, and the translated version.

Afterwards, depending on the input being ASH or ASCII, I would transmit the ASH code via a different colour via LED. To do this, I globally declared the relative units of time ASH required. I would turn on the LED, delay for the appropriate unit of time. Then turn off the LED and delay again before the next blink. The red LED flashes for ASCII input, and blue for ASH input.

To set the delays for the length of time an LED needed to flash, I first declared their base units of time globally. Then by reading a value from the potentiometer, I could multiply its value by the base units of time so that by adjusting the value of the potentiometer, the length of the transmission changed. The potentiometer has a maximum value of 1023. I wanted the maximum value for transmission to be 500ms for 1 unit of time. So to make so that the potentiometer had a linear change in transmission speed from 0 to 500, I divided the value by 2.046, as $1023/2.046 = 500$. This meant that I could also have more control over the exact timing of the transmission, as every 2 degrees of turn now is equal to 1 degree of turn before.

I believe that I have completed all of part 1, as well as all of part 2. I did not complete or attempt part 3. I have not found any issues or bugs with my code yet, and find it to be perfectly functional.

As for the mark on the assignment, I would award myself a 2.1 or a 2.2. I think this score is fair as I think I have shown through my code and comments a strong understanding of the problem given. I believe I have used correct grammar throughout and make good use of the technology employed with correct terminology usage. As far as functionality goes I have omitted some of the more

advanced pieces or work, which makes me feel as though I could dip down to a 2.2, as if the reason I am sitting on the fence as to what mark I would give.