MIT AppInventor 2

Na tentativa de comunicar o Arduino com algum SmartDevice, foram escolhidos o SmartPhone e a conexão bluetooth.

Para a conexão bluetooth no arduino, foi usado o módulo HC-05. Para gerenciar a conexão estabelecida entre o módulo e o Smartphone, foram desenvolvidos: um programa em uma linguagem baseada na linguagem C na plataforma arduino e um app android desenvolvido através da plataforma MIT AppInventor 2.

Com o objetivo de conseguir controlar 4 servomotores remotamente, fui gradualmente incrementando a complexidade de um circuito, com o objetivo de novamente me familiarizar com as peças e a programação.

Acender led com botão; Piscar 4 leds; Controlar qual dos 4 leds deve acender de acordo com o posicionamento do joystick; Variar a intensidade dos 4 leds de acordo com o posicionamento do joystick; Aplicar o conceito aprendido nos projetos anteriores para controlar um servomotor; Acender um led com o módulo bluetooth HC-05; Utilizar o módulo bluetooth HC-05 para controlar a luminosidade de um led; Utilizar o módulo bluetooth HC-05 para controlar a luminosidade de 4 leds; Aplicar os últimos avanços para o controle de 4 servomotores.

Cada uma das etapas serviu para consolidar cada vez mais o conhecimento previamente adquirido, conseguindo então manter um desenvolvimento constante e gradual.

Uma das etapas na qual mais foi despendido tempo, foi a tentativa de ligar 2 leds (que representam o controle de 2 servomotores) através da conexão bluetooth.

Após refletir sobre as maneiras de se diferir qual led acender e em qual luminosidade deve ser acendido (sendo uma representação de qual servomotor deve ser girado e em que angulação ele deve girar), optei por seguir duas possibilidades:  
 A primeira sendo o envio de um número de 4 algarismos, sendo o primeiro algarismo (algarismo na casa do milhar) variando entre 1 e 4 e os 3 últimos, variando de 0 a 255 que são os valores possíveis de iluminação dos LEDs (ou no caso dos motores de 0 a 180). Sendo assim, aqui vão alguns exemplos:  
o valor 1127 acenderia o led 1 com a intensidade 127,  
o valor 2012 acenderia o led 2 com a intensidade 12,  
o valor 3255 acenderia o led 3 com a intensidade 255 (completamente ligado),  
o valor 4000 acenderia o led 4 com a intensidade 0, ou seja, desligaria o led 4.  
 E a segunda sendo o envio de um texto de 4 caracteres, sendo o primeiro caractere variando entre ”a”, ”b”, ”c” e ”d” e os 3 últimos, variando de 0 a 255 que são os valores possíveis de iluminação dos LEDs (ou no caso dos motores de 0 a 180). Sendo assim, aqui vão alguns exemplos:  
o texto “a127” acenderia o led “a” com a intensidade 127,  
o texto “b012” acenderia o led “b” com a intensidade 12,  
o texto “c255” acenderia o led “c” com a intensidade 255 (completamente ligado),  
o texto “d000” acenderia o led “d” com a intensidade 0, ou seja, desligaria o led “d”.

Até a etapa anterior (ligar apenas 1 led), era possível enviar o valor de 1 byte (8 bits, ou seja, era possível enviar números de 0 até 255) do app para o arduino sem maiores problemas, porém, isso é justamente a gama de valores necessários para o controle da intensidade de um, e apenas um led em sua completude de valores.

Ao se acrescentar mais um, dois ou três leds seriam necessários ao menos 510 valores (para o controle de 2 leds, o que ultrapassa o máximo alcançado por 1 byte. Se eu tentasse enviar algum número superior a 255 usando apenas 1 byte, a informação era perdida. Analisando através do monitor serial dava pra ver que o número recebido não era o mesmo que o enviado.

A plataforma MIT AppInventor 2, permite que os dados sejam enviados em forma de 1 byte, 2 bytes, 4 bytes, lista de bytes e texto.

Pela lógica, se 1 byte conseguia tranquilamente enviar a informação de 1 led, então 2 bytes deveriam suprir tranquilamente a necessidade proveniente do uso de 2 leds, porém aí veio um novo problema: Os 2 bytes estavam sendo enviados separadamente, fazendo com que, ao invés de receber um grande número (que utilize 2 bytes), eu recebesse dois números de 1 byte e que eu não estava conseguindo reagrupar de forma alguma.

Fiz um App de testes apenas para analisar como que é efetuada a transmissão e visualização de dados entre os Apps feitos pelo AppInventor e os códigos do Arduino. Assim, foi possível chegar a conclusão de que a melhor alternativa seria a de enviar como texto a informação a ser passada, e no código do Arduino, manipular de acordo com o necessário para que o texto se tornasse “utilizável”.

Assim consegui, através do aplicativo, manipular dois leds, o que pode ser facilmente expansível para 4 leds, e posteriormente para 4 servomotores.

Com isso, pude perceber um bug: Ao mudar a intensidade luminosa muito rapidamente, é como se os vários valores enviados acumulassem, e isso muda o valor enviado.