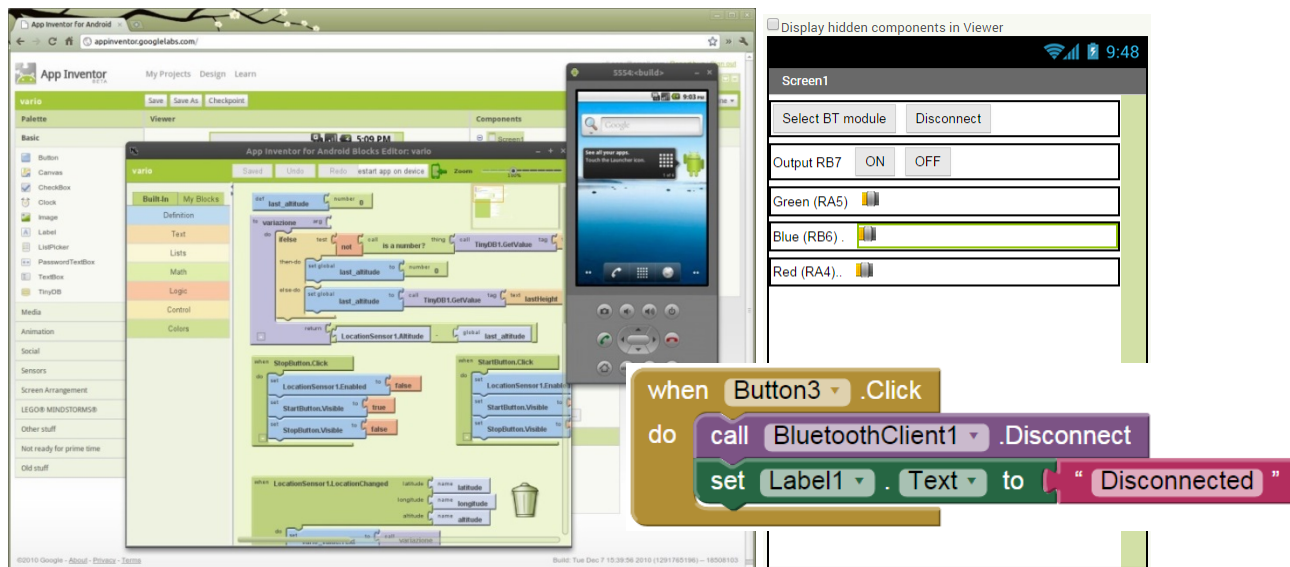


APPINVENTOR



APPINVENTOR is een 100% gratis online tool waarmee je zelf apps kan maken voor Android devices (niet voor Apple).

De open politiek van Android maakt dat het ontwikkelen van apps voor Android devices vrij eenvoudig en kosteloos kan verlopen.

Link: <http://appinventor.mit.edu/explore/index-2.html>

>> Volg deze “get started” lessen om je eerste stappen te zetten in APPINVENTOR

<http://appinventor.mit.edu/explore/get-started.html>

>> bekijk de vele lessen en instructiefilms die beschikbaar zijn om deze omgeving te leren kennen.

>> Klik op “create apps” om je eigen app te maken

!! aanloggen via appinventor kan enkel met een google account

>> Installeer de “AI2 companion” APP op uw smartdevice en test uw app live uit.

CONNECTIE VIA BLUETOOTH


De meeste SMARTDEVICES hebben een ingebouwde BLUETOOTH module. APPINVENTOR geeft ons de kans om op een eenvoudige manier data te verzenden en te ontvangen via die ingebouwde BLUETOOTH module van onze SMARTDEVICES. 'So far so good' dus.

De volgende hindernis is om deze draadloze BLUETOOTH signalen op een zo eenvoudig mogelijke manier in te lezen in onze microcontroller. Microcontrollers hebben geen ingebouwde BLUETOOTH module aan boord. We moeten hiervoor dus een connectie maken met een externe BLUETOOTH module. Hiervoor hebben wij de keuze gemaakt voor de HC-06 module.

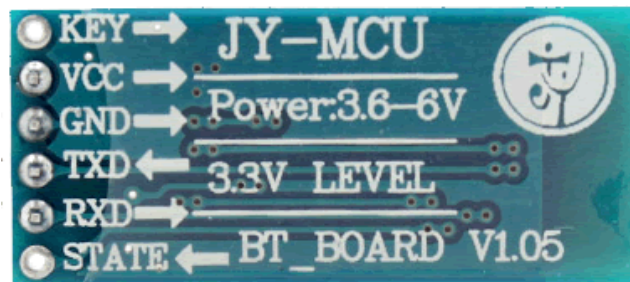
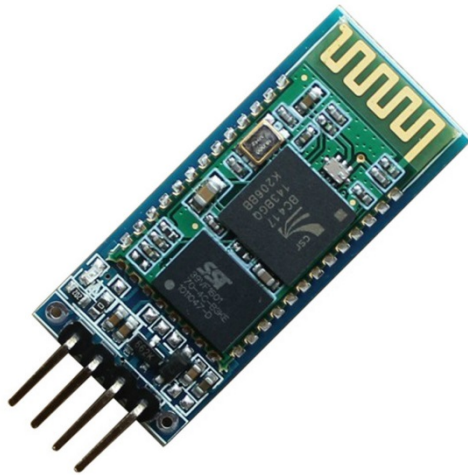
Deze HC06 modules zetten BLUETOOTH signalen om naar RS232 signalen en omgekeerd. Die RS232 signalen kunnen we perfect inlezen in onze microcontrollers. Moest u hier meer informatie over willen: de volledige RS232 communicatie is behandeld in Videolessen deel 3.

Hier ziet u de standaard instellingen, maar die zijn via AT commando's over RS232 te veranderen. De datasheet zal je hierbij ondersteunen. Wij gebruiken de standaard instellingen en hebben dus geen AT commando's nodig.

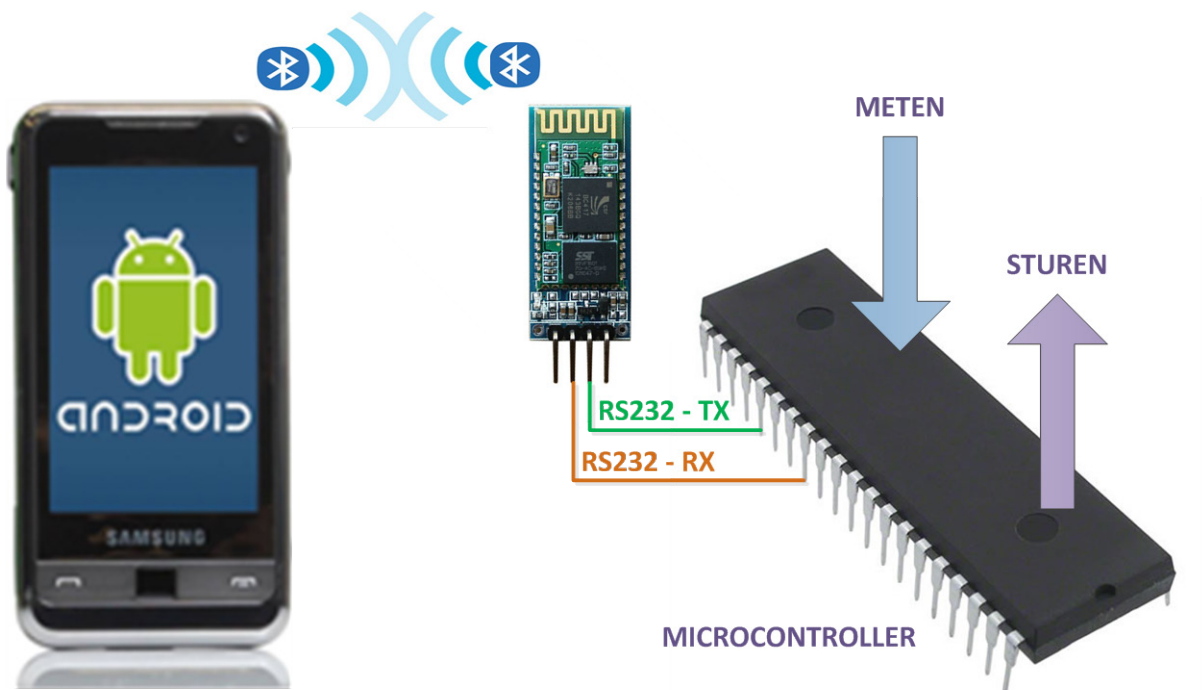
- Full duplex RS232 via TX en RX
- Baud Rate: 9600 bps (standaard – maar kan tot 115200 gaan)
- Data : 8 bits
- Stop Bits: 1 bit
- Parity : None
- Handshake: None
- BLUETOOTH Passkey: 1234

Deze HC-06 BLUETOOTH module is een "Class 2" BLUETOOTH device en kan daardoor normaal gebruikt worden tot afstanden van maximaal 10 meter.

Je ziet op deze HC-06 module de twee BLUETOOTH IC's zitten, een PCB antenne en een aantal externe randcomponenten. Het nadeel van deze module is dat deze – door zijn vorm - niet rechtsreeks bruikbaar is in zelfbouw projecten en dat deze HC-06 module op 3.3V werkt terwijl de meeste van onze microcontrollers op 5V werken.



Daarom kopen wij onze HC-06 modules rechtsreeks op een moeder-PCB. Deze moederprint is voorzien van een voltage aanpassing die voedingsspanningen tussen 3.6 en 6V toelaat en een 4 pin connector met VCC, GND, en een TX en RX pin.



Om even samen te vatten: We willen via een zelfgeschreven APP op ons SMARTDEVICE - bepaalde actuatoren die aan onze microcontroller hangen - kunnen bedienen. En van bepaalde sensoren die aan onze microcontroller hangen willen we de metingen op ons SMARTDEVICE kunnen tonen.

Als communicatiesysteem gebruiken we langs de kant van het SMARTDEVICE de ingebouwde BLUETOOTH module en langs de kant van de microcontroller gebruiken we de HC-06 module die BLUETOOTH signalen omvormt naar RS232 signalen.

Deze lessen zijn zo opgebouwd dat we in een aantal stappen gaan leren om eerst actuatoren aan te sturen via APPINVENTOR en dan om sensoren in te lezen in APPINVENTOR.

We behandelen in deze lessen enkel wat nodig is om te meten en te sturen via BLUETOOTH. Naar alle waarschijnlijkheid gebruiken we hiervoor amper 10% van de mogelijkheden van APPINVENTOR. Laat u dus vooral niet tegenhouden om zelf op ontdekking te gaan.

INTERRUPT

Om er zeker van te zijn dat onze microcontroller nooit inkomende data mist werken we hier met een interrupt. We gebruiken de "USART1_RX_vect" interrupt die is ingebouwd in deze microcontroller en die het programma meteen onderbreekt als er nieuwe data ontvangen wordt via de RX pin van UART1 (waarop onze HC-06 module is aangesloten)

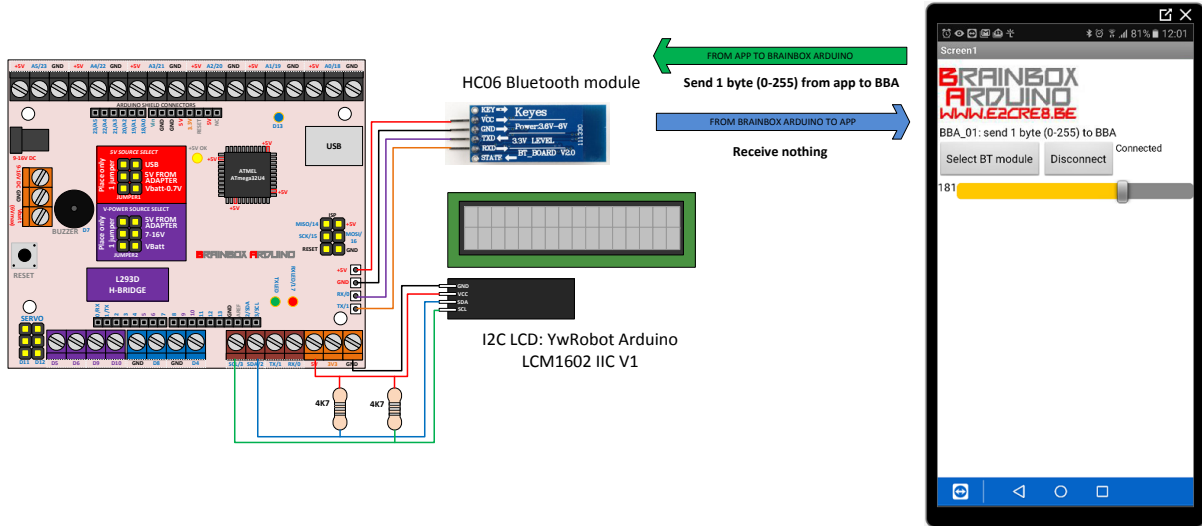
```
ISR (USART1_RX_vect)    // interrupt routine that executes every time a new byte is received
{
    //char ReceivedByte;
    ReceivedByte = UDR1; // Fetch the received byte value into the variable " ByteReceived " - the RXint flag is automatically cleared now
    state = !state;      // inverse the variable "state"
}
```

Op dat moment worden de instructies in de interrupt routine onderaan het programma doorlopen alvorens verder te gaan met het hoofdprogramma. In dit geval lezen we in deze interruptroutine dus één byte in en inverteren we de waarde van de variabele 'state' (om les 13 aan te sturen)

Interrupts worden onder Arduino IDE best geconfigureerd in echte C code en de echte registernamen zoals u in dit programma kan zien. Voor alle ander instructies mag u gewoon de uitgebreide Arduino libraries gebruiken.

PROGRAMMA 1 "BBA_01"

Hardware:



- Connecteer de HC06 Bluetooth module correct aan de BBA
- Connecteer de I2C LCD correct aan de BBA

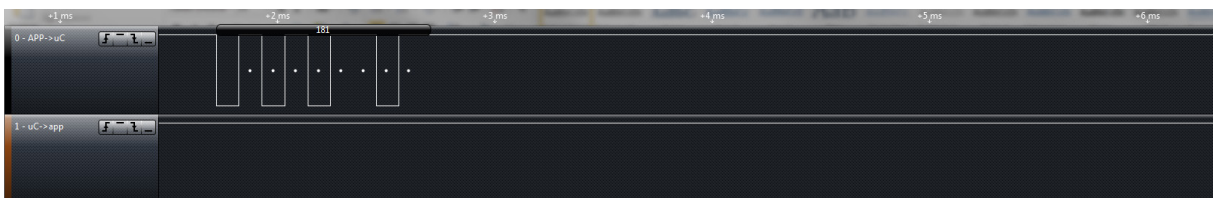
Software:

- Download het programma BBA_01 via Arduino IDE naar de BBA. Op het LCD scherm verschijnt "TEST BT HC06 BBA_01"
- Open de file BBA_01.aia in appinventor en connecteer je Smartphone met Appinventor om deze app uit te testen
- Of... installeer de BBA_01.apk file op uw Smartphone en test deze app uit

>> druk op "Select bluetooth module"

!!Let wel op: Hiervoor moet de BLUETOOTH functie van uw SMARTDEVICE wel staan ingeschakeld en er moet in de eigenschappen reeds een connectie gelegd zijn met dit HC-06 BLUETOOTH device.

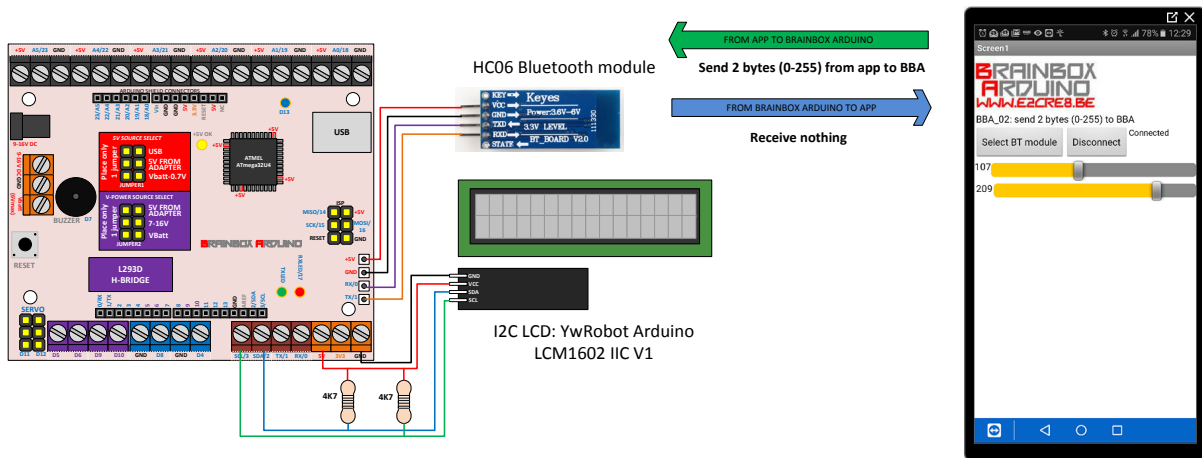
>> schuif met de slider op uw app – de waarde – tussen 0 en 255 – verschijnt nu op de LCD
 >> telkens als de app een nieuwe waarde stuurt, zal de uC de interruptroutine uitvoeren en zal de led13 op de BBA aan of uitgaan.



Dit Logic analyser beeld laat mooi zien hoe de byte data van de APP naar de uC wordt gestuurd.

PROGRAMMA 2 "BBA_02"

Hardware:



- Connecteer de HC06 Bluetooth module correct aan de BBA
- Connecteer de I2C LCD correct aan de BBA

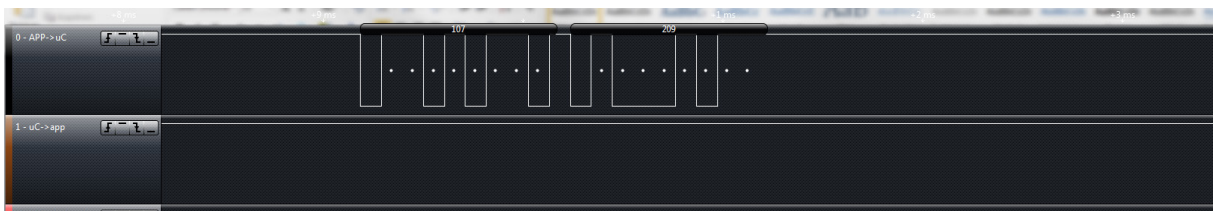
Software:

- Download het programma BBA_02 via Arduino IDE naar de BBA. Op het LCD scherm verschijnt "TEST BT BBA_02"
- Open de file BBA_02.aia in appinventor en connecteer je Smartphone met Appinventor om deze app uit te testen
- Of... installeer de BBA_02.apk file op uw Smartphone en test deze app uit

>> druk op "Select bluetooth module" – wacht tot de tekst 'connected' verschijnt.

!!Let wel op: Hiervoor moet de BLUETOOTH functie van uw SMARTDEVICE wel staan ingeschakeld en er moet in de eigenschappen reeds een connectie gelegd zijn met dit HC-06 BLUETOOTH device.

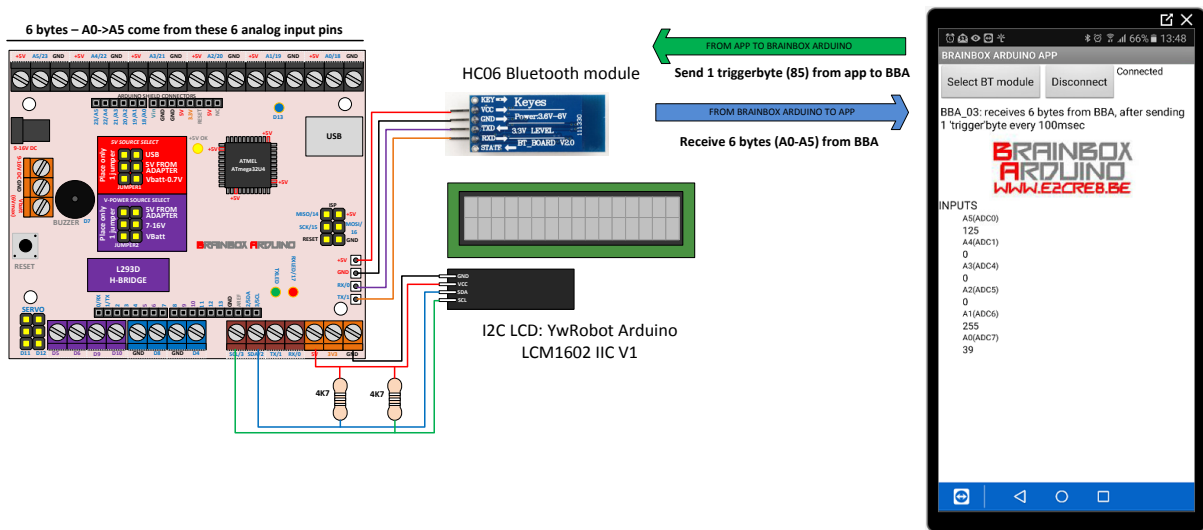
>> schuif met de 2 sliders op uw app – de 2 waarden – tussen 0 en 255 – verschijnen nu op de LCD
 >> telkens als de app een nieuwe waarde stuurt, zal de uC de interruptroutine uitvoeren en zal de led13 op de BBA aan of uitgaan.



Dit Logic analyser beeld laat mooi zien hoe de 2 bytes data van de APP naar de uC wordt gestuurd.

PROGRAMMA 3 "BBA_03"

Hardware:



- Connecteer de HC06 Bluetooth module correct aan de BBA
- Connecteer de I2C LCD correct aan de BBA
- Connecteer 6 sensoren of schakelaars aan pins A0->A5 zoals voorgeschreven in de werkbladen over de betreffende sensoren. Je mag een sensoringang ook aan 5V (leest als 255) of aan GND hangen (leest als 0)

Software:

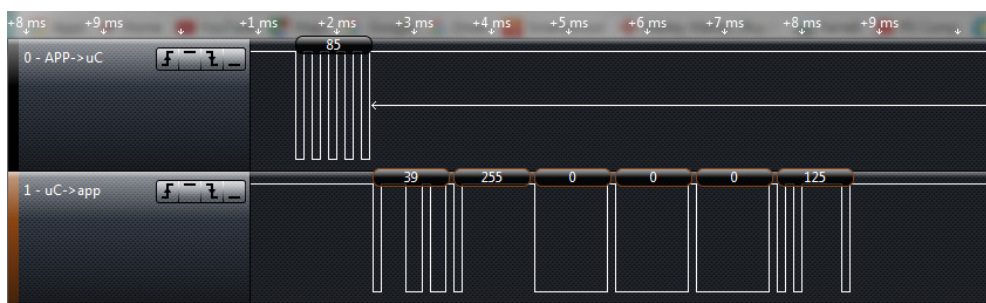
- Download het programma BBA_03 via Arduino IDE naar de BBA. Op het LCD scherm verschijnt "TEST BT BBA_03"
- Open de file BBA_03.aia in appinventor en connecteer je Smartphone met Appinventor om deze app uit te testen
- Of... installeer de BBA_03.apk file op uw Smartphone en test deze app uit

>> druk op "Select bluetooth module" – wacht tot de tekst 'connected' verschijnt.

!!Let wel op: Hiervoor moet de BLUETOOTH functie van uw SMARTDEVICE wel staan ingeschakeld en er moet in de eigenschappen reeds een connectie gelegd zijn met dit HC-06 BLUETOOTH device.

>> Elke 100 msec stuurt de app een byte (met de willekeurig gekozen waarde 85) naar de BBA om nieuwe meetwaarden op te vragen.

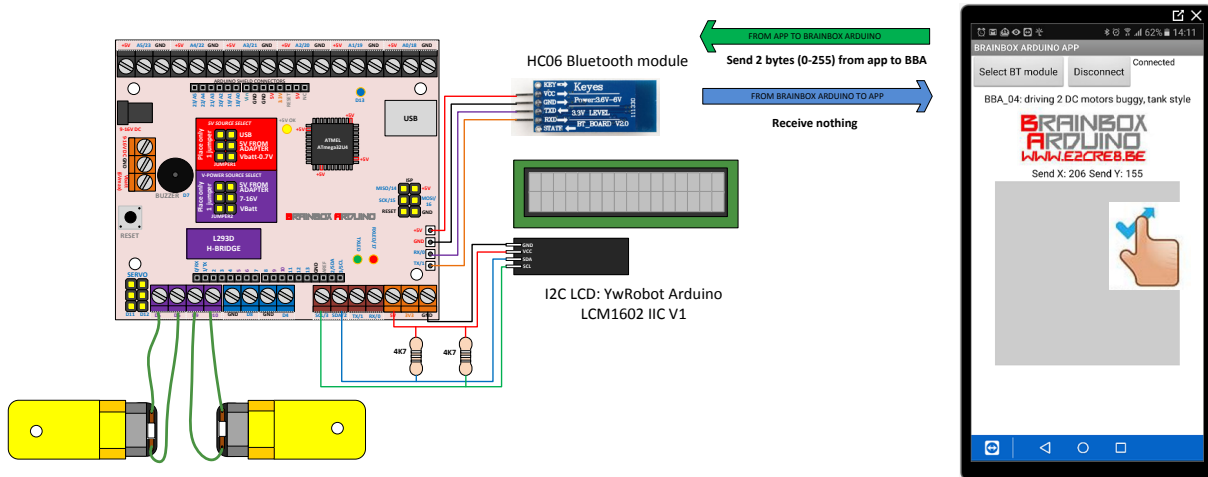
>> De BBA antwoord meteen door de 6 nieuwe meetwaarden door te sturen naar de APP.



U ziet hier mooi de triggerbyte (85) en de 6 bytes als antwoord die van de BBA naar de APP gezonden worden.

PROGRAMMA 4 "BBA_04"

Hardware:



- Connecteer de HC06 Bluetooth module correct aan de BBA
- Connecteer de I2C LCD correct aan de BBA
- Connecteer 2 DC motoren aan de BBA. **!! De draairichting van de motoren wordt bepaald door de aansluiting. Test de app, als de draairichting niet correct is moet je de draden van die betreffende motor omdraaien.**

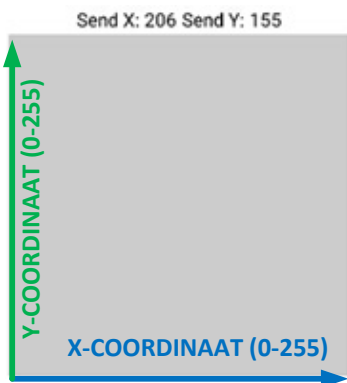
Software:

- Download het programma BBA_04 via Arduino IDE naar de BBA. Op het LCD scherm verschijnt "TEST BT BBA_04"
- Open de file BBA_04.aia in Appinventor en connecteer je Smartphone met Appinventor om deze app uit te testen
- Of... installeer de BBA_04.apk file op uw Smartphone en test deze app uit

>> druk op "Select bluetooth module" – wacht tot de tekst 'connected' verschijnt.

!!Let wel op: Hiervoor moet de BLUETOOTH functie van uw SMARTDEVICE wel staan ingeschakeld en er moet in de eigenschappen reeds een connectie gelegd zijn met dit HC-06 BLUETOOTH device.

>> Wanneer er met de vinger over het grijze vierkant (het canvas) geslept wordt zullen er 2 nieuwe bytes, met respectievelijk een X en een Y coördinaat naar de BBA gestuurd worden.



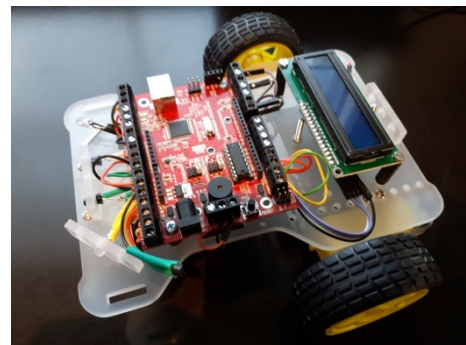
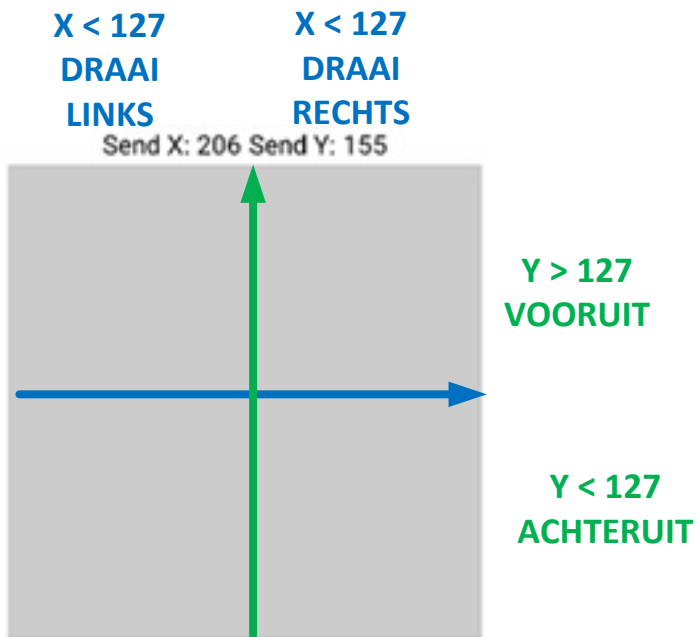
In de BBA wordt dit X en Y coördinaat vertaald naar 4 variabelen.

```
int DC_LEFT = 0; // om de power (Duty Cycle PWM) naar de linker motor te bepalen

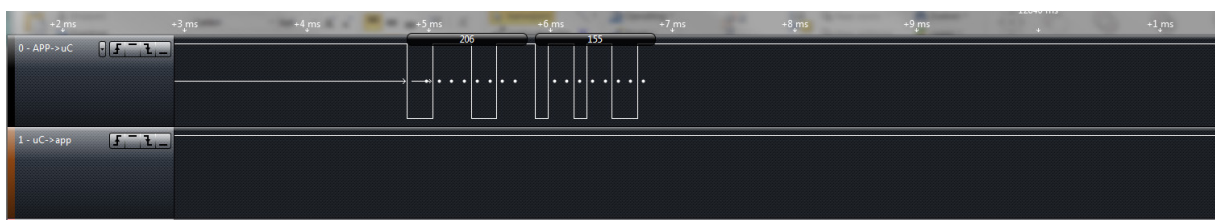
byte DIR_LEFT = 0; // om de draairichting naar de linker motor te bepalen (0 = vooruit, 1 = achteruit)

int DC_RIGHT = 0; // om de power (Duty Cycle PWM) naar de rechter motor te bepalen

byte DIR_RIGHT = 0; // om de draairichting naar de rechter motor te bepalen (0 = vooruit, 1 = achteruit)
```



Door uw vinger over het canvas te bewegen kan je de BBA, opgebouwd op de Buggy, met de batterijen aangesloten en met de jumpers op VBatt-0.7 en Vbatt, besturen in alle richtingen.



U ziet hier mooi dat er slechts 2 bytes verzonden worden van de app naar de BBA.