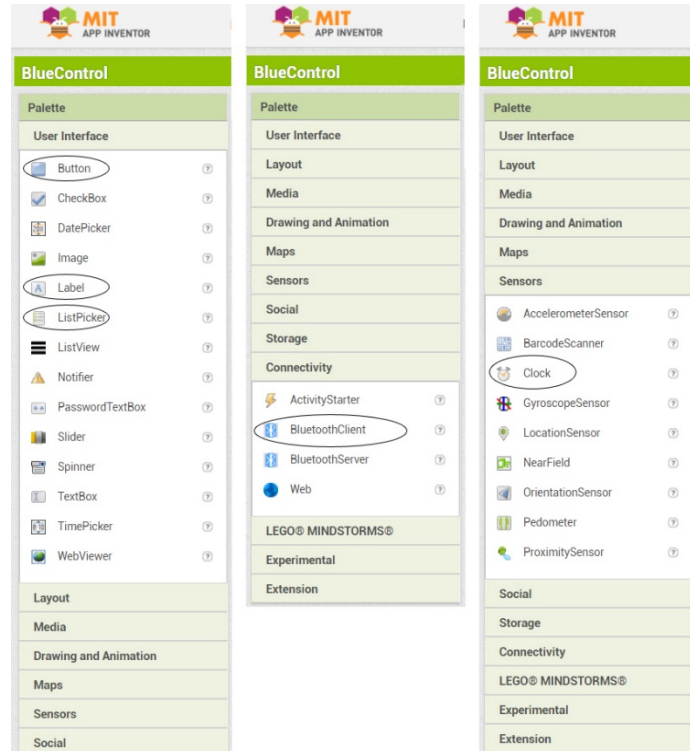


## Roboter mit dem Handy steuern

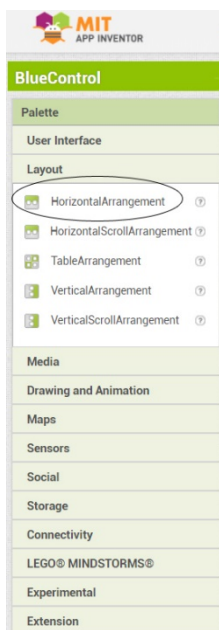
### Appdesign

Um einen Roboter mit dem Handy zu steuern, wird eine App benötigt, die auf einfache Weise mit dem „App-Inventor 2“ realisiert werden kann. Dazu muss das Programm im Internet gestartet und aus der „Palette“ die benötigten Komponenten auf den „Viewer“ bzw. den „Screen“ (virtueller Bildschirm) gezogen werden. Der Reihe nach sind das:

1. Ein „ListPicker“, der bluetoothfähige Geräte erkennen und in einer Liste anzeigen kann, aus der das gewünschte ausgewählt werden muss.
2. Ein „Label“, das anzeigen kann, ob eine Verbindung zum gewählten Bluetooth-Modul hergestellt worden ist oder nicht.
3. „Button(s)“, die Steuerbefehle an den Roboter übermitteln.
4. Ein „BluetoothClient“, der die Verbindung zwischen den Geräten überwacht.
5. Üblich ist auch, eine Uhr mit einzubinden.



Als Steuerbefehle sind ▲ vorwärts, ▼ rückwärts, ◀ links, ▶ rechts und **STOPP** vorgesehen, die wie folgt zusammengestellt werden sollen:



Um die Knöpfe „links“, „rechts“ und „STOPP“ nebeneinanderzulegen, muss in der „Palette“ unter „Layout“ das Feld „Horizontal(Scroll)Arrangement“ auf „Screen1“ platziert werden, in das die „Buttons“ der Reihe nach eingefügt werden. Die „Buttons“ werden über „Properties“ und „Image“ mit passende Grafiken ausgestattet.

Mit Hilfe leerer „TextBox(es)“ an geeigneten Stellen fungieren, wenn unter „Properties“ die Schriftgröße („Height“ und „Width“) variiert wird.

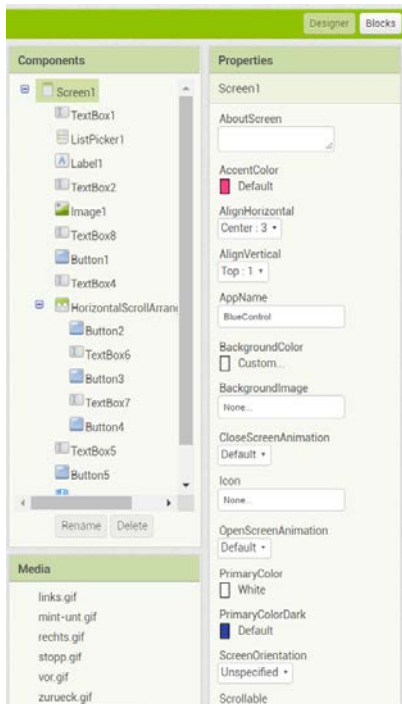
Weitere Grafiken können bei Bedarf über „Palette“, „User-Interfaces“ und „Image“ eingefügt werden.

Rechts das „Design“ im „Viewer“ nach Fertigstellung.



## Programmierung der Komponenten

Für die Programmierung im App-Inventor rechts oben auf „Blocks“ klicken und dann in der linken



Spalte unter „Components“ und „Screen1“ den Eintrag „ListPicker1“ anwählen. In dem sich öffnenden Auswahlménú den braunen Block „when ListPicker1.BeforePicking – do“ anklicken. Durch weitere Klicks den dunkelgrünen Block „set ListPicker1.Elements to“ auswählen und in die Lücke des ersten



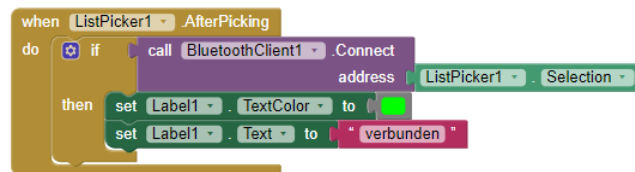
Blocks hinter „do“ einfügen. Der noch fehlende hellgrüne Block „BluetoothClient1.AddressesAndNames“ findet sich unter „BluetoothClient1“ und wird entsprechend der Abbildung oben angefügt.

Dieser Teil der Programmierung sorgt dafür, dass die verfügbaren Bluetoothgeräte angezeigt werden.

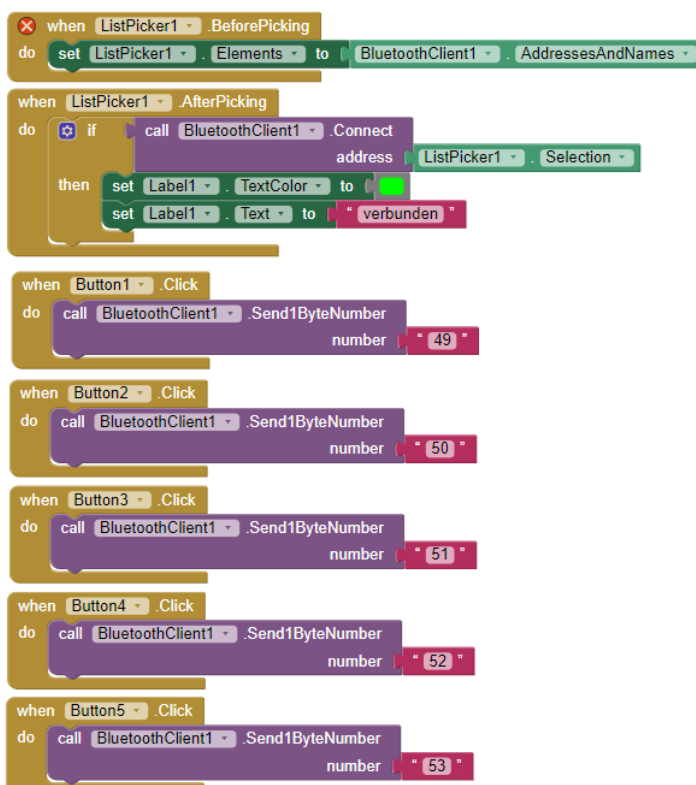
Ein weiterer Programmteil muss dann für dessen Einbindung sorgen.

Dazu durch einen Klick unter „Components“ auf „ListPicker1“ das Auswahlménú öffnen und den braunen Block „when ListPicker1.AfterPicking – do“ auswählen. Danach unter

„Components“, „Built-in“, „Control“ den Block „if – then“ auswählen und in die Lücke des vorherigen Blocks hinter „do“ einfügen. Hinter „if“ wird dann der lila Block „call BluetoothClient1.Connect – address“ eingefügt usw.



Die weiteren Schritte zur Fertigstellung der Programmierung sollten nach diesen Erklärungen mithilfe der folgenden Abbildung selbst gefunden werden können im Vergleich mit der folgenden



Abbildung, die den kompletten Code umfasst.

Die Zahlen 50 bis 53 entsprechen den Tasten 1 bis 4 auf dem Computer. Das ist bedeutsam, um die Zahlenangaben im Arduino-Quellcode zu verstehen.

Ist die Programmierung fertiggestellt, wird sie zur Ausführung auf dem Smartphone gespeichert. Dazu im App-Inventor ganz oben auf „Build“ klicken und in dem sich öffnenden Menü „App (save.apk to my computer)“ auswählen.

## Arduino-Programmierung

Der Arduino, der das Fernsteuerfahrzeug überwacht, muss nun so programmiert werden, dass er auf die von der Smartphone-App über Bluetooth ausgegebenen seriellen Steuersignale empfangen und darauf reagieren kann. Hier ein tauglicher Code:

```
int senden; //Variable "senden" einbinden
void setup() {
  pinMode(6, OUTPUT); //Anschlusspins der Motoren festlegen
  pinMode(9, OUTPUT);
  pinMode(10, OUTPUT);
  pinMode(11, OUTPUT);

  Serial.begin(9600); //Seriellen Monitor einbinden
}
void loop() {
  if (Serial.available())
  {
    senden = Serial.read();
  }
  if (senden == '1') //Wenn eine 1 von der App gesendet wird ...
  {
    analogWrite(6, HIGH); //... schalte beide Motoren auf Vorlauf
    digitalWrite(9, LOW);
    analogWrite(10, HIGH);
    digitalWrite(11, LOW);
  }
  if (senden == '2') // Wenn eine 2 gesendet wird ...
  {
    analogWrite(6, HIGH); //... schalte den linken Motor ab
    digitalWrite(9, LOW);
    digitalWrite(10, LOW);
    digitalWrite(11, LOW);
  }
  if (senden == '3') //Wenn eine 3 gesendet wird ...
  {
    digitalWrite(6, LOW); //... schalte beide Motoren ab
    digitalWrite(9, LOW);
    digitalWrite(10, LOW);
    digitalWrite(11, LOW);
  }
  if (senden == '4') //Wenn eine 4 gesendet wird ...
  {
    digitalWrite(6, LOW); //... schalte den rechten Motor ab
    digitalWrite(9, LOW);
    analogWrite(10, HIGH);
    digitalWrite(11, LOW);
  }
  if (senden == '5') //Wenn eine 5 gesendet wird ...
  {
    digitalWrite(6, LOW); //... schalte beide Motoren auf Rücklauf
    analogWrite(9, HIGH);
    digitalWrite(10, LOW);
    analogWrite(11, HIGH);
  }
}
```