

Heute zeige ich Ihnen, wie man eine App für ein Androidgerät, Handy oder Tablett, ganz einfach mit Hilfe grafischer Elemente entwickeln kann, um damit zum Beispiel das Robot Car zu steuern. Herzlich willkommen zur neuen Folge von

Micropython auf dem Robot Car Teil 3 - Wir steuern das Auto mit dem Handy

Das Handy erkennt die Lage im Raum und stellt zum Beispiel die Bildschirmdarstellung je nach Ausrichtung des Geräts automatisch um. Dahinter steckt ein Verwandter des MPU6050-Chips, der auf dem Modul GY521 verbaut ist, das wir in der ersten Folge in unsere Handsteuerung eingebaut haben. Vielleicht haben Sie dieses Feature bereits in der zweiten Folge zusammen mit Ihrem Robot Car ausprobiert, was rede ich, natürlich haben Sie das schon getan. Nun, was mit der Handsteuerung funktioniert, können Sie nach Durchführung der Schritte in dieser Beschreibung auch mit Ihrem Handy machen. Keine Sorge, es wird nicht kompliziert. Wenn Sie in jungen Jahren gerne mit Bauklötzen gespielt haben, meistern Sie das problemlos.

Besorgen und Installation der Software

Die Hardware ist im Prinzip komplett, es sei denn sie wollen Ihrem Auto noch zwei LEDs für die Rücklichter spendieren. Was Sie jetzt und hier brauchen, sind demnach nur zwei Dinge, das Robot Car und Ihr Android-Smartphone. Na ja, und dann ist da noch die Software. Um die freie Software vom MIT (Massachusetts Institute of Technology), das auch Lizenzinhaber ist, zu nutzen, gibt es drei Wege. Zwei davon führen über Softwareteile, die auf dem PC installiert werden müssen. Ich habe die dritte Möglichkeit gewählt, die auch von den Machern von App-Inventor 2 (AI2) vordringlich empfohlen wird. Auf dem PC kommt dazu nur ein Browser zum Einsatz und auf dem Handy die App MIT Al2 Companion. Auf dem PC tut's ein Browser wie Firefox oder Chrome. Mit Internet Explorer funktioniert es definitiv nicht, aber der ist eh tot. Ob Edge einsetzbar ist, habe ich nicht getestet, wozu auch? Den MIT Al2 Companion können Sie über den Google Playstore installieren. Damit diese Vorgehensweise korrekt arbeitet muss das Handy in einem lokalen Funknetz eingeloggt sein, nicht über das Internet. Ein Zugriff über das Internet kann nicht funktionieren, weil vom Handy aus zwar beliebige Ziele erreichbar sind, umgekehrt das Handy aber vom Internet aus nicht direkt angesprochen werden kann. Es ist im lokalen Netz des Providers gekapselt und deshalb von außen nicht sichtbar. Das heißt, der nötige Zugriff des Browsers auf das Handy aus dem Internet ist nicht möglich, weil keine öffentliche IP und kein Routing existieren.

Das grundsätzliche Werkzeug steht nun bereit. Es gibt aber ein kleines Handicap, das Handy kann zwar perfekt den TCP-Dialekt, versteht aber erst einmal kein Wort UDP. Das ist für die Standardanwendungen auch nicht erforderlich. Also verpassen wir ihm via Al2 einen kleinen Sprachkurs. Den habe ich auf Ullis Roboterseite gefunden. Hier ist die Downloadadresse:

https://ullisroboterseite.de/android-AI2-UDP/UrsAI2UDP.zip

Entpacken Sie das Archiv zunächst in ein beliebiges Verzeichnis. Wechseln Sie dort hin und überzeugen Sie sich, dass die Datei **de.ullisroboterseite.ursai2udp.aix** aufgelistet ist. Sie enthält die Erweiterungen, die wir zur Erzeugung unserer Steuer-App in Kürze brauchen.

App Inventor 2 erstmals starten

Öffnen Sie jetzt Chrome oder Firefox, falls noch kein Browser läuft und geben Sie folgende URL ein.

http://ai2.appinventor.mit.edu

Die Anmeldung bei Appinventor ist nur über ein Googlekonto möglich. Haben Sie bereits eines, dann klicken Sie auf **Weiter**, sonst auf **Konto erstellen**.

	Anmelo	lung
	Weiter zu n	nit.edu
E-Mail oder T	elefonnummer	
E-Mail-Adres	se vergessen?	
Wenn Sie fort	fahren möchten, m	üssen Sie zustimmen,
dass Google Spracheinste	ihren Namen, Ihre E Ilung und Ihr Profilb	-Mail-Adresse, Ihre bild an mit.edu weiter gibt.
	3	5
		101-11

Das Konto kann auch mit gefakten Daten angelegt werden. Das gilt für die erste und auch für die folgende Seite.

Jens	Mustermann	
Nutzername		
jemus95	@gmail.com	
Passwort	Bestätigen	nur einem Konto.
	•••••	
8 oder mehr Zeichen mi und Symbolen verwende	t einer Mischung aus Buchstaben, Ziffern en	

VIIKOI	nmen bei Goog	le	
jemus9	5@gmail.com		
— -	Telefonnummer (optiona	al)	
/ir verwende icht für ande	n Ihre Nummer zum Schutz Ihre ere sichtbar.	es Kontos. Sie ist	•
E-Mail-Adı	resse zur Kontowiederhers	tellung (option	
Damit schü	tzen wir Ihr Konto		
Tag		Jahr	line and the second second
07	Monat -	1991	sind bei uns sicher und geschützt
Ihr Geburts	datum		
Geschlecht -	te dies nicht beantworten	*	
varum wir	nach diesen Informatione	en Tragen	

Nach den üblichen Einstellungen zur Privatsphäre landen Sie auf dieser Seite.



Der MIT APP INVENTOR -Link führt Sie zurück zum Login. Melden Sie sich mit Ihrem Googlekonto an und lassen Sie sich nicht von dem Riesenangebot an Tutorials und

Beispielen verführen, sondern starten Sie einfach ein neues, leeres Projekt, Sie wollen doch das Robot Car mit einer eigenen App steuern, oder? Natürlich wollen Sie das!

	GET STARTED WITH SOME TUTORIALS	S
HELLO PURR	TALK TO ME	TRANSLATE APP
		դո
HelloPurr is a simple app that you can build in a very short time. You create a button that has a picture of a cat on it, and then program the button so that when it is clicked a "meow" sound plays.	Text to Speech is surprisingly fun. Find out for yourself with this starter app that talks.	Quickly translate English to Spanish (and other languages too!) You're challenged with creating an app that could act as an aid for immigrant parents who need a little extra help in English-speaking situations. Inspired by YR Media story What It's Like to be a Translator.
GO TO TUTORIAL	GO TO TUTORIAL	GO TO TUTORIAL

Ich weiß aber auch, dass Sie später begierig alles aufsaugen werden, was Al2 Ihnen bietet - mir ging es jedenfalls so.

Create new App Invent	or project	
Project name:	robotcar	
Cancel	OK	

Einrichtung der Bildschirmobjekte

Werfen wir einen Blick auf die Seite auf der das Layout der App entworfen wird. In dem Abschnitt **Palette** finden Sie die Zutaten. Der mittlere Bereich, **Viewer**, ist Ihre Arbeitsfläche. Rechts davon in der Sektion **Components** wird die Hierarchie der verwendeten Komponenten dargestellt und rechts daneben, in **Properties**, die Eigenschaften der gerade ausgewählten Komponente. Das alles sehen Sie, wenn sie in der **grünen Zeile** über diesen Bereichen rechts außen **Designer** ausgewählt haben.



Wir werden jetzt Stück für Stück zum Layout hinzufügen und die Eigenschaften der Komponenten einstellen. Alle Eigenschaften, die nicht explizit aufgeführt sind, bleiben auf Defaulteinstellung.

Die Screenkomponente ist bereits aktiviert. Wir geben ihr den Titel ROBOTCAR STEUERUNG. Scrollen Sie in Properties nach unten und geben Sie bei Title den Text ein.

tle
OBOTCAR STEUERUNG

In der ersten Zeile sollen ein Schalter und ein Textfenster erscheinen. Also holen Sie aus dem Ordner **User Interface** einen **Switch** und ein **Label** und ziehen sie die beiden auf die Handyfläche. Was, das Label will nicht neben dem Switch wohnen. Geben Sie den beiden halt ein ordentliches Zuhause, holen Sie sich aus dem Ordner **Layout** ein **HorizontalArrangement** und setzen Sie es über den Schalter. Jetzt können Sie die Elemente Switch1 und Label1 hineinsetzen und schauen Sie, jetzt nehmen sie auch friedlich neben einander Platz.

Lay	vout	
	HorizontalArrangement	0
	HorizontalScrollArrangemen	t 🔊
	TableArrangement	?
8	VerticalArrangement	?
	VerticalScrollArrangement	(7)



Verteilen wir die Eigenschaften. Markieren Sie in Components den Eintrag HorizontalArrangement1. Klicken Sie auf das Feld von Width, Fill Parent und OK.

AlignHoriz	ontal	
Left:1 🔹		
lignVertio	al	
Top:1 •		
Backgrour Defaul	idColor t	
leight		
10 percent.		
Width		
Automa	ntic	
Fill pare	ent	
Q	pixels	
)	percent	
Canaal	OK	

Markieren Sie in **Components Switch1**. Linksklick auf **Rename**, dann geben Sie den neuen Namen **MainSwitch** ein. Ändern Sie die Eigenschaften wie folgt.

Tex	d .
RU	N
Tex	rtColor
	Default
Th	umbColorActive
	Default
Th	umbColorInactive
	Dark Gray
Tra	ckColorActive
	Default
Tra	ckColorInactive
	Red
Vis	ible
\square	

Benennen Sie Label1 in MyIP um und stellen Sie die Eigenschaften auf folgende Werte ein.

Fill parent	
Text	
None	.4
FextAlignment center : 1 🔹	
left : 0	
center : 1	

Fügen Sie zwischen **MainSwitch** und **MyIP** ein weiteres Label mit dem Namen **nameMyIP** und dem Text **Meine IP** ein.

Das nächste Arrangement heißt **ConfigIP**, ist vertical, der Hintergrund gelb, die Höhe 20% des Displays und es nimmt die gesamte Bildschirmbreite ein. Ich denke, Sie kriegen das hin!?

Display hidden components in Viewer Phone size (505,320)	 Screen1 HorizontalArrangement1
• •	MainSwitch
ROBOTCAR STEUERUNG	ConfigIP

Jetzt zur Einrichtung. Ein **Spinner** ist eine Dropdownliste, aus der Einträge durch Touch ausgewählt werden können. Die Einträge werden durch eine Liste festgelegt. Die Begriffe sind durch Kommas getrennt. Beim Bildschirmaufbau ist der erste Begriff bereits markiert. Eine **Textbox** ist ein Eingabefeld, dessen Inhalt beim Verlassen übernommen wird.

Wir legen eine Liste von IP-Adressen fest, die mögliche Ziele des Projekts darstellen, WLAN-Accesspoint, ESP32-Accesspoint und PC zu Testzwecken. Die Liste sieht aus wie folgt: Klar, dass Sie die IP-Adressen an Ihre Netzumgebung anpassen müssen.

Default, 10.0.1.101, 10.0.2.101, 10.0.1.10

Die ausgewählte Adresse soll zur Laufzeit in der Textbox dargestellt werden, um die Auswahl mitzuteilen und /oder um Änderungen zu ermöglichen.

Die Eigenschaften des Spinners sind hier dargestellt.



Finden Sie die Eigenschaften der Textbox **txtFreeIP** heraus? Die Schriftgröße verrate ich Ihnen: 18.

		😪 🕼 📓 9:48
ROBOTCAR STEUERUNG		
RUN 🛑 Meine IP:	None	
add items		
d	efault	
	elault	

Haben Sie alles? BackgroundColor: Orange, FontSize: 18, Width: Fill Parent, Text: default, TextAlignment: center 1, TextColor: Blue.

Es folgt jetzt ein **TableArrangement** mit 2 Spalten und 2 Zeilen. Ich nenne es DriveData. Es füllt die ganze Bildschirmbreite und enthält 4 Lables, die zwei linken, **txtVelo** und **txtDir**, zur Beschreibung und die rechten, **IbIVelo** und **IbIDir**, für die Wertangabe von Geschwindigkeit und Fahrtrichtung.

txtVelo und txtDir: FontSize: 14, Textalignment: right 2, Width: 50% Text: Geschwindigkeit | Richtung

IblVelo und IblDir: FontSize: 20, FontBold, Text: 0

Im nächsten Arrangement (horizontal, height: 20%, Width: Fill Parent) finden zwei Buttons Platz, btnLight und btnMotor. Für die Buttons gilt: BackgroundColor: Green, FontSize: 30, Shape: rounded, TextAlignment: center 1, Text LICHT | MOTOR.

In einem anschließenden HorizontalArrangement, VeloMaxDisplay (Width: Fill Parent), wohnen ein Beschreibungslabel, txtVeloMax (Width: 50%, TextAlign: center 1, Text: Maximale Geschwindigkeit) und ein Label IbIVeloMax, das den gerundeten Wert des folgenden Schiebereglers darstellt.

Den Abschluss bildet ein Slider (aka Schieberegler) mit Namen sldVeloMax mit folgenden Properties:

sldVeloMax	
ColorLeft Red	
ColorRight Cyan	
Width	
Fill parent	
MaxValue	
25	
MinValue	
10	
ThumbEnabled	
ThumbPosition	
10	
Visible	

Unser Display sieht nun so (ähnlich) aus.

		9:48 🖻 🗐
ROBOTCAR S	TEUERUNG	
RUN 🛑	Meine IP:	None
add items		~
	default	
Gi	eschwindigkeit () Richtung ()	
LIC	снт мо	TOR
Maximale Ges	chwindigkeit 3	
\bigtriangledown	0	

Jetzt fehlen uns noch zwei "unsichtbare" Elemente, ein Accelerometer und der UDP-Sender. Den Accelerometer Sensor finden Sie im Palette-Ordner **Sensors**. Ziehen Sie ihn auf die Handyfläche. Er wird unterhalb derselben angezeigt.

robo	tcar	S
Palet	te	
Search	Components	
Use	er Interface	
Lay	out	
Me	dia	
Dra	wing and Animation	
Maj	ps	
Sen	ISOFS	
۲	AccelerometerSensor	2
	BarcodeScanner	3
Ŧ	Barometer	9

Wissen Sie noch in welchem Ordner sich die Datei

de.ullisroboterseite.ursai2udp.aix befindet? Diese Datei brauchen wir als Nächstes. Wir müssen damit den Sprachumfang von Al2 erweitern. Rollen Sie in **Palette** ganz nach unten, klicken Sie dort auf **Extension** und dann auf **Import extension**.

robotcar	S
Palette	
Search Components	
User Interface	
Layout	
Media	
Drawing and Animation	
Maps	
Sensors	
Social	
Storage	
Connectivity	
LEGO® MINDSTORMS®	
Experimental	
Extension	
Import extension	

Im folgenden Fenster navigieren Sie nach Klick auf **Durchsuchen** zur aix-Datei.

Import an extension into project	
From my computer	URL
Durchsuchen	roboterseite.ursai2udp.aix
Cancel	Import
Cuncer	mpore

Nach Klick auf **Import** werden nach ein paar Sekunden zwei Einträge angezeigt. Ziehen Sie UDPXmitter auf die Handyfläche. Jetzt beherrscht Ihr Handy auch den UDP-Dialekt und kann Datagramme an den Server auf dem Robot Car senden. Wenn Sie dieses Feature auch in anderen Projekten nutzen möchten, müssen Sie den Importvorgang jedes Mal wiederholen, nachdem das neueProjekt angelegt wurde.

Die Programmierung

Wenden wir uns nun der Programmierung zu. Bis jetzt haben wir ja nur Daten und Objekte bereitgestellt. Jetzt erwecken wir das Ganze zum Leben, jetzt dürfen Sie Bausteine stapeln. Wechseln Sie in der grünen Menüleiste mit dem Button **Blocks** rechts außen ins Programmierfenster von Al2.

Links ist der Bereich für Programmblöcke oder Bausteine mit eingebauten Blocks für Text, Logik, Datenstrukturen, mathematischen Funktionen und Operatoren, Variablen und Proceduren. Darunter finden Sie die Blocks der Elemente, die wir bisher festgelegt haben.

Im Viewerfenster bauen wir jetzt für verschiedene Events aus den Bausteinen unsere "Häuser". Sie können nur Bausteine zusammenstecken, die auch zusammenpassen. Das entspricht den Syntax- und Semantikregeln von MicroPython und weiteren Programmiersprachen. Blocks wie in Al2 sind Sie vielleicht schon mal begegnet. Tools wie Squeak und Scratch basieren auf der gleichen Methode zu programmieren. Dahinter steckt die Sprache Smalltalk-80. Zugegeben, die Sache ist etwas holprig für Leute, die gewohnt sind, Programmtexte frei weg zu schreiben. Aber mal ehrlich, wie oft vertippen Sie sich und suchen dann nach den Fehlern? Wenn Fehler in Scratch oder Al2 auftreten, dann sind das logische Fehler, aber keine Syntaxfehler. Daher ist diese Art zu programmieren für Anfänger gut geeignet und wird in Schulen gern praktiziert.

Wir bereiten schon mal verschiedene Variablen vor, dann initialisieren wir den Viewport. Ich werde zu Beginn alles sehr genau beschreiben. Im weiteren Verlauf werden dann die Angaben spärlicher und beschränken sich schließlich nur noch auf wesentliche Dinge wie die Auflistung der Properties. Die Farben der verschiedenen Bereiche helfen bei der Orientierung. Unbenannte Strukturblöcke stammen aus dem Built-In-Bereich, die Beschriftung der benannten Blöcke verrät ihre Herkunft.

Ziehen Sie aus Variables ein initialize global name to in den Viewer. Hängen Sie aus Math eine 0 dran. Klicken Sie auf name und geben Sie xw ein, Enter. Widerholen Sie den Vorgang für yw.

initialize global name to a
set D to N
initialize local name to the in
initialize local name to a in the second sec

Erzeugen Sie eine globale Variable **targetIP** und hängen Sie aus Text ein " " dran. Geben Sie dann die IP-Adresse Ihres PCs ein. Zur Demonstration des Einfügens und Verklebens von Bausteinen sollte das genügen. Wir ergänzen später weitere Variablen



Ziehen Sie jetzt aus Screen1 eine when Screen1 initialize-Klammer.

🚫 se	t UDPXmitter1 🔹 Remot	teHost 🔹	to
	Duplicate		
	Add Comment		
	Collapse Block		
	Disable Block		
	Add to Backpack (9)		
	Delete Block		
	Make Generic	1	
	Download Blocks as PNG	(\tilde{r})	
	Help		
	Dolt		

Ein Rechtsklick auf den set UDPXmitter1-Streifen aus **UDPXMitter1** und ein Klick auf **Duplicate** erzeugen ein Duplikat des Blocks. Verkleben Sie den zweiten mit dem ersten und Ändern Sie im zweiten Block RemoteHost in **RemotePort**. Aus Variables holen Sie ein **get** und aus Math eine 0. Kleben Sie das get an RemoteHost und die 0 an RemotePort. Ändern Sie die 0 in 9000 und im get-Block wählen Sie targetIP aus.

Aus **MyIP** holen Sie ein **set MyIP.Text to** und kleben Sie ein **join** aus Text dran. Kleben Sie aus UDPXmitter1 ein **LocalHost** und **LocalPort** ans **join**. Für den trennenden Doppelpunkt brauchen Sie eine weitere Kontaktstelle. Ein Klick auf das blaue Symbol öffnet das Fenster für die Einstellung. Ziehen Sie **string** aus der linken Hälfe zwischen die beiden in der rechten Hälfte und ergänzen Sie dann den Doppelpunkt.



Wenn Sie keine Fehlermeldungen und Warnungen haben, sind Sie sicher gespannt darauf, wie das alles auf Ihrem Handy aussieht. Verbinden Sie das Handy mit dem WLAN-Router und starten Sie auf dem Handy den Al2-Companion.



Besonders einfach geht die Kontaktaufnahme zwischen Handy und PC über den QR-Code-Scan. Auf PC-Seite starten Sie den Vorgang mit Klick auf **Connect** und **Al-Companion**.

Screen1	·I	Al Companion Emulator	ublish to Gallery
Viewer		USB	
whe	n E	Refresh Companion Screen	
do	se	Reset Connection	eHost 🔹 to 🚺 get global targetIP 🔹
	se	Hard Reset	ePort 🔹 to 🌾 9000
	set	MyIP . Text . to	🖸 join 📔 UDPXmitter1 🔹 . LocalHost 🔹

Scannen Sie jetzt den QR-Code mit dem Handy ein.

robotcar Progress Bar		
	15%	
15 Rendezvous (please wait)		
	Dismiss	

Der Progressbar läuft durch und danach haben Sie Ihren Entwurf mit dem ersten Programmierergebnis auf dem Handydisplay. Die lokale IP des Smartphones wird rechts neben Meine IP angezeigt. Sie können auch schon den Schalter antippen, die Spinnerliste aufklappen und den Schieberegler bedienen. Damit dadurch auch Aktionen ausgelöst werden, müssen wir noch eine Menge Code ergänzen. Übrigens, nicht erschrecken, die Verbindung zum Handy wird relativ schnell wieder gekappt, wenn keine Aktionen erfolgen. Dadurch wird nichts gelöscht oder verändert, Sie müssen nur wieder neu verbinden. Wenn Sie später die komplette App auf dem Handy installieren, gibt es auch kein automatisches Trennen der Verbindung mehr.

So bringen Sie Leben in die Buttons für das Fahr-Licht und das Motor-Relais.



Wie die Namen der when-Klammern erkennen lassen, stammen sie aus dem Blockvorrat der Buttons btnLight und btnMotor. die Functioncalls stammen aus UDPXmitter1. Die Kommandostrings müssen mit einem Linefeed abgeschlossen werden, das erledigt das "\n".

Jetzt geht es dem Spinner an den Kragen. Für die Behandlung der Auswahl erzeugen wir eine lokale Variable **IPselect** und weisen aus dem **selection**-Feld **get selection** zu.



Der Rest der "Füllung" erklärt sich von selbst.



Für die Ermittlung der Fahrstufen brauchen wir eine Funktion, welche die Floating-Rohwerte vom Accelerometer in brauchbare Integerwerte im zulässigen Bereich umwandelt. Zum Einstellen der Bereichsobergrenze dient der Slider. Damit man den eingestellten Wert auch vor Augen hat, weise ich **IbIVeloMax.Text** den gerundeten Wert des Schiebereglers zu. Für die Berechnungen brauche ich zwei neue globale Variablen.



Für die Berechnung der Fahrstufen benötige ich das Verhältnis aus der maximal möglichen Fahrstufe und den maximalen Absolutwerten des Accelerometerchips. Diese habe ich zu 9 in alle Richtungen gemessen. Der berechnete Wert steht in **VeloFaktor** bereit.

So wie die Rohwerte vom Accelerometer kommen, sind die Vorzeichen für beide Achsen komplementär. In der Funktion **procedure2** wird daher nach der Einführung von einigen lokalen Variablen das Signum **vz** des Parameters x berechnet und, falls ungleich Null, invertiert.

Die Fahrstufe wert wird nach folgender Formel berechnet:

Für die Rückgabe des Ergebnisses dient die neue globale Variable Result. Damit das in einem Rutsch geht definieren wir auch gleich noch zwei Merker für xw und yw aus dem vorherigen Durchgang.



get X
e (-1 × () get in ▼ / (absolute ▼) get in ▼)
0
r get vz ▼ × (floor ▼) ♥ (♥ r get global VeloFaktor ▼) × (absolute ▼) get (n ▼) + (0.5)
o f if f absolute t get wert t ≥ t 2
then 🖡 get wert 🕐
else (0

Damit sind wir auch schon beim letzten Block angelangt. Hier werden die Acceleratorwerte erfasst, auf Fahrstufen umgerechnet und falls sich ein Wert geändert hat, wird er an das Robot Car gesendet. Damit das alles auch wirklich passiert, muss der Schalter **MainSwitch** natürlich an sein.

when Acc	elerometerSensor1 💿 AccelerationChanged
xAccel	yAccel zAccel
do 🔯 if	MainSwitch 🔹 On 🔹
then	call procedure2 •
	x (get xAccel •)
	set global xw * to (get global Result *)
	call procedure2 •
	x get vAccel
	set global yw v to I get global Result v
	set IblVelo . Text to get global yw
	set [blDir]. Text I to get global xw
	If get global xw ▼ I get global xwOld ▼
	then set global yw to i get global yw
	call UDPXmitter1 .XmitAsync
	Message 🛛 🙆 join (" 🔽 "
	get global yw 🔹
2	
	ger giobal xw
	then set global xwOld to get global xw T
	call UDPXmitter1 .XmitAsync
	Message 🔅 join 🖸
	get global xw *

Jetzt ist alles im Kasten und damit ist es an der Zeit, den finalen Funktionstest am Handy zu absolvieren. Starten Sie das Robot Car und verbinden Sie Handy und PC, wie Sie es oben schon mal gemacht haben. Mainswitch an, Handy nach vorne neigen und los geht's. Fahren Sie ein paar Kurven, wenden Sie – perfekt!

Dann lassen Sie uns aus dem Projektentwurf eine dauerhafte App machen, die Sie auf dem Handy installieren können. Compiliert wird die Anwendung in jedem Fall in der AI-Cloud. Dort wird Ihr Projekt auch gespeichert und von dort wird die apk-Datei heruntergeladen. Der Button **Build** bietet zwei Möglichkeiten an. Mit der ersten Option wird nach ca. 1 bis 2 Minuten nach dem Anklicken ein QR-Code angezeigt. Er enthält die Download-URL. Scannen Sie ihn ein. Nach dem Download startet die Installation. Bestätigen Sie die Sicherheitsabfragen alle mit 'ja'



Die zweite Option bietet nach der Übersetzung den Download auf Ihren PC an. Speichern Sie die apk-Datei in einem beliebigen Verzeichnis und transferieren Sie sie danach mit Bluetooth oder einer anderen Methode (e-Mail, WWW, ...) auf Ihr Handy. robotcar.apk (apk = Android PacKage) landet vermutlich im Downloadordner und der liegt in 'Eigene Dateien'. Prüfen Sie zunächst, ob Setups aus diesem Ordner durchgeführt werden dürfen.

Einstellungen ...



Eigene Dateien ...

13:07 🔛 🖼 🚥 🔹	19. jil 🗎	13:07 🖼 🖬 🚥 🔹	™ , d ≜	13:07 🖾 🖬 🖬 🔹	Se 🗎	
Eigene Dateien	a 🧖	<	a 🗉 🤗	< 0	ג 🗉 🧖	
Aktuelle Dateien		Interner Speicher		▶ Interner Speicher ▶ Down	load	
Katagorian		Alarms		28.05.2020 15:02	7,90 KB	
	P	01.01.2019 13:32	0 Objekte	manage_ESP-7.apk 27.08.2020 20:59	3,32 MB	
Bilder Videos	Audio	Android 09.05.2020 07:49	3 Objekte	manage_ESP.apk	2.22 MD	
Dokumente Downloads	APK Installationsdateie n	data 09.05.2020 02:47	1 Elemente	Micropython_esp.pdf	3,32 MD	robotcar
Interner Speicher		DCIM 09.02 21:18	2 Objekte	_ 22.09.2020 10:14	486 KB	Möchtest du diese App installieren?
 18,43 GB / 32,00 GB SD-Karte 094 GB / 59,45 GB 		Download 22.03 11:57	12 Objekte	UDP_Transmitter.apk	3,30 MB	Abbrechen Installieren
en Sie sich an, was Speicherolatz a	if Ihrem Telefon beleat.	Movies 01.01.2019 13:32	0 Objekte	UDP_Transmitter(1).apl	k 3.30 MB	
🔇 Speicher ana	ysieren	Music 01.01.2019 13:32	0 Objekte	UDP_Transmitter(2).apl	k 330 MB	robotcar
		Notifications 01.01.2019 13:32	0 Objekte	UDP_Transr (3).aph	k	
		D :-+		20.03 10:40 V	3,30 MB	Fertig Öffnen

Nach dem finalen Test sollten Sie die Erlaubnis, dass Apps aus dem Ordner 'Eigene Dateien' installiert werden dürfen, aus Sicherheitsgründen widerrufen.

Eine Übersicht der Programmblöcke können Sie als <u>PDF</u> herunterladen. Auch die Projektdatei <u>robotcar.aia ist als Download</u> verfügbar, ebenso die <u>Setupdatei</u> <u>robotcar.apk</u>. Ein Listing gibt es für dieses Projekt leider nicht.

Ich wünsche Ihnen viel Freude am weiteren Experimentieren mit Micropython und dem Robot Car. Erweitern sie doch die Handysteuerung mit der Freigabe des Starts für das autonome Fahren Ihres e-Autos. Natürlich müssen dafür noch Routinen, zum Beispiel für das Folgen einer Spurlinie, geschrieben werden. Mit dem Hinzufügen eines weiteren Tasters zur ESP32-Handsteuerung lässt sich das Gleiche dort auch umsetzen. Oder wie wäre es mit einer automatischen Garagentorsteuerung via Infrarot-Modulen bei Annäherung des Autos? Der ESP32 bietet dazu alle Voraussetzungen.

Catch the challenge!

Lin	klis	te:

Teil1	HTML-Format	
Teil2	HTML-Format	
Teil1	deutsches PDF	
	englisches PDF	
Teil2	deutsches PDF	
	englisches PDF	
Teil3	deutsches PDF	
	englisches PDF	