

#### ANLEITUNG ZUM AUFBAU EINES MOBILEN TEMEPERATUR-REGELUNGSVERSUCHS

13.03.2018, Version 1 Sebastian Fischer





Nr	Bezeichnung	Anzahl	Preis	Link
1	Arduino Due R3 SAM3X8E 32-bit ARM Cortex-M Control	1	13,42	Arduino Due
	Board Module +Cable New		€	
2	Protoshield + Steckbrett	1	6,29€	<u>Shield</u>
3	LM35cz Temperatur Sensor	1	3,35€	LM35cz
4	USB auf Micro USB Kabel	1	1,15€	USB / Micro USB
5	Micro USB port	1	3,97€	Micro USB Breakboard
6	Leistungswiderstand Panasonic ERG-2SJ100 10R2W	1	0,35€	<u>Widerstand</u>
7	4,7 K Widerstand	1	0,10€	
8	1,2 K Widerstand	1	0,10€	
9	Gehäuse	1	21€	<u>Gehäuse</u>
10	Verbindungskabel	1	3,89€	<u>Kabel</u>
11	MOSFET IRL 540N	1	0,70€	Mosfet irl 540N
12	Ventilator	1	14,03€	<u>Ventilator</u>
13	Matlab 2018a Lizenz	1	-	
14	SIMULINK Support Package for Arduino Hardware	1	-	
15	Simulink Support Package for Arduino Hardware	1	-	
16	Embedded Coder (Version 7.0)	1	-	
17	System Identification Toolbox (Version 9.8)	1	-	
18	MATLAB Support Package for Arduino Hardware (Version	1	-	
	18.1.1)			
19	MATLAB Support for MinGW-w64 C/C++ Compiler	1	-	
	(Version 18.1.0)			
20	Curve Fitting Toolbox (Version 3.5.7)	1	-	
21	Bioinformatics Toolbox (4.10)	1	-	
22	Uebertragungsfunktion.slx	1	-	
23	Projektarbeit2_Steuerung.slx	1	-	
24	Projektarbeit2_SteuerungMitGW.slx	1	-	
25	Projektarbeit2_Regelung.slx	1	-	
26	Projektarbeit2_DisturbanceCompensation.slx	1	-	
27	Frequenzgang_Messung.slx	1	-	
28	StartConditions_Tf.m	1	-	
29	Bodeplot.m	1	-	
30	Execute_FreqRespMeas.m	1	-	
31	Exectue_Transferfunction.m	1	-	
32	FrequResponse.m	1	-	
33	Plots.m	1	-	
34	Transfcn.m	1	-	
Summe			68,3€	

Anbei eine Stückliste aller Teile, Simulink Modelle und Matlab Files inkl. Kauf-Link:



Falls Unklarheiten bei Hardware Aufbau auftreten, kann sich immer an folgenden Schaltplan gehalten werden:

fritzing

## Steckbrett



Die Verbindungen zwischen den Steckbrettlöchern verlaufen, in dieser Ansicht vertikal. Die grünen Punkte veranschaulichen dies zum besseren Verständnis. Löcher, welche sich horizontal voneinander befinden, sind nicht miteinander verbunden (siehe rote Punkte). Zudem ist die obere Hälfte des Steckbrettes nicht mit der unteren Hälfte verbunden.

#### 2





Zu aller erst muss die Folie auf der Rückseite des Steckbretts entfernt werden und das Steckbrett anschließend auf dem Protoshield wie in der Abbildung dargestellt draufgeklebt werden.







Hier sei darauf zu achten, dass der Mosfet richtig herum auf das Steckbrett gesteckt wird. Sprich die Fläche mit dem Loch ist hinten und die dunkle Fläche mit der Beschriftung vorne. Anschließend den Mosfet noch um ca. 90° nach hinten biegen, damit dieser später ins Gehäuse passt.



Die "Beine" des 1,2 k $\Omega$  Widerstand (Farbcode: braun – rot – rot) müssen zunächst mit einem Seitenschneider gekürzt und umgebogen werden bevor er an gezeigter Stelle platziert wird.



Auch die Beine des 4,7 k $\Omega$  Widerstandes (Farbcode: gelb – violett – schwarz – braun) müssen gekürzt und gebogen werden, bevor dieser an gezeigter Stelle aufgesteckt wird.



Für den Lastwiderstand gilt dasselbe wie für die Widerstände in Punkt 4 und 5.





Die Wölbung des Temperatursensors LM35cz muss nach außen zeigen und die Ebene Fläche zum Widerstand.













Der "Ground" Fuß des Temperatursensors muss entsprechend mit dem Ground Pin des Protoshields verbunden werden. Der zweite und dritte Pin von rechts sind beides GND Pins und somit frei wählbar.

## Schritt 10



Der mittlere "Fuß" des Temperatursensors wird mit dem Analog Pin 2 (dritter von links) verbunden.



Der rechte "Fuß" des Temperatursensors wird mit dem 5V Pin des Protoshields verbunden (vierter Pin von rechts).

### Schritt 12



10















Schritt 15





















Das Protoshield so auf den Arduino Due aufsetzen, sodass die jeweiligen Pins zueinander passen und fest aufstecken.







Der Temperatursensor wird nun mittels Kabelbinder an den Widerstand gedrückt. Dazu den Kabelbinder unter dem Widerstand durchführen und um den Temperatursensor festziehen. Abschließend noch den Kabelbinder mittels Seitenschneider kürzen.



Das Gehäuse wird gemäß mitgelieferter Aufbauanleitung zusammengebaut, jedoch werden die 2 dargestellten Teile nicht verwendet.



Den fertigen Hardware Aufbau in das Gehäuse legen, sodass die Arduino micro USB Anschlüsse zu den Ausschnitten des Gehäuses zeigen.



Das blaue Kabel des Arduino Due wird nun durch den Gehäuse Ausschnitt durchgeführt und mit dem unterem USB Port des Arduino Due verbunden.

Das andere Kabel wird durch den anderen Gehäuseausschnitt durchgeführt und mit dem auf dem Steckbrett befindlichen micro USB Port verbunden

Zu guter Letzt wird nur noch der Deckel auf das Gehäuse geschraubt und fertig ist der Hardware Aufbau.



Fertiger Hardware Aufbau an den Laptop USB Ports angeschlossen.

#### Softwareeinrichtung

Nachdem der Hardware Aufbau abgeschlossen ist, müssen alle nötigen MATLAB Skripte und Simulink Modelle auf dem Computer abgespeichert werden. Z.B. können alle Dateien im Matlab Verzeichnis unter einem bestimmten Ordner abgespeichert werden. Dieser Ordner muss dann in Matlab ausgewählt werden. Dazu muss der Ordner wie links im Bild mit der rechten Maustaste angeklickt werden, anschließend "Add to Path" und "Select Folders and Subfolders" auswählen. In meinem Beispiel heißt der Ordner in dem die Dateien abgelegt sind "Projektarbeit2".

4	MA	TLAB R2	018a -	academic use										
	Н	HOME PLOTS APPS					DITOR		PUBL	.ISH	VIEW			
	ŀ			G Find Files	<ul> <li></li></ul>	l Com	nsert 📘 ment ᠀	∫ <i>fx</i> 6 ‰	<b>₩</b> ▼	0 0   <sub>1</sub> ,1 1	D	> 🖳		
N	lew T	Open	Save	🛁 Print 👻	🔍 Find 👻	Ir	ndent 📑		1	Breakpoi	nts Ri	In Run and Advance		
			FILE	_	NAVIGATE		E	DIT	_	BREAKPOIN	ITS			
4	•	1		C: Users	Sebastian +	Docum	ents 🕨	MATLA	AB ►					
Cu	urren	t Folder				$\odot$	📝 Ed	itor - C	:\Users\	Sebastian	Documer	nts\MATLAB\Pro		
		Name 4	<u>م</u>				+18	Exec	cute_Ste	uerung.m	× Exe	cute_Regelung		
±		Add-On	s				1		% load	d('DisT	K2_2_n4	4_ohne(2).		
±		Archiv					2		% load	d('DisT	K2_2_n4	4(2).mat')		
±		Example	es eboit1				3		% load	d('DisT	K2_5_n4	1.mat')		
	ñ.	Projekta	rbeit2				4		% load	d('DisT	K2_05_1	14(2).mat'		
÷	1	pzMove	C	Open	Eingab	e 5	-	d = 0.	.1; % L	iniendi	icke Plots			
±		Sensors_	S	Show in Explore	er		6		W = 2;					
±		slprj GM2PID run_GM2					8.	_	Δ2m =	τ π2(20)	•%Tempe	erature Re		
			C	Lreate Zip File		50	9.	_	A20 =	т(20);	%Temper	rature Rea		
				kename Delete		FZ	10 -	-	A05 =	T05(20	);%Temp	erature R		
				Jelete		Losche	n 11 -		A5 = 1	INeu (20	);%Temp	perature R		
				New Folder			12					_		
				vew rile			13 ·	-	E2m =	T2 (end	);%Temp	perature_R		
			0	Compare Selec	ted Files/Folders		14 -	-	E2o =	T(end)	;%Tempe	erature_Rea		
			C	Compare Agair	nst		>15 -	-	E05 =	т05 (en	d);Temp	perature_R		
			C	Tut		Stra+X	16 -	-	E5 = 1	INeu (en	d);%Ter	nperature_]		
			0	ζοργ		Stra+C	17							
			P	Paste		Stra+V	omn	nand V	Vindow					
			-			b g · ·	<u> </u>	>				1		
			A	Add to Path		>	Selecte	ted Folders						
			× 1	ndicate Files N	ot on Path			Selecte	ed Folde	rs and Sub	ofolders			
			F	Refresh		F5								

Nun müssen noch diverse Matlab Add-Ons installiert werden. Dazu öffnet man Matlab und wählt im Reiter "Home"  $\rightarrow$  "Environment" die Schaltfläche Add-Ons aus. Anschließend nach dem gewünschten Package suchen und installieren.

📣 MAT	🛦 MATLAB R2017a - academic use														
НС	OME		PLOTS	APPS	s	EDITOR	PUB	LISH	VIEW						
New Script	New	Open	Find Files	Import Data	Save Workspace	New Variab	le ble ▼ space ▼	Analy & Analy	ze Code and Time Commands 🔻	Simulink	Layout	<ul> <li>⊘ Preferences</li> <li>Set Path</li> <li>Parallel ▼</li> </ul>	Add-Ons	? Help ▼	Community
		FILE			V	ARIABLE			CODE	SIMULINK		ENVIRONMENT			RESOURCES

Folgende Add-Ons müssen installiert werden:

- Simulink Support Package for Arduino Hardware
- Curve Fitting Toolbox (Version 3.5.7)

- System Identification Toolbox (Version 9.8)
- MATLAB Support Package for Arduino Hardware (Version 18.1.1)
- MATLAB Support for MinGW-w64 C/C++ Compiler (Version 18.1.0)
- Embedded Coder (Version 7.0)
- Bioinformatics Toolbox (4.10)

Außerdem muss noch ein MATLAB file ausgetauscht werden, damit der Widerstand nicht weitergeheizt wird nachdem das Simulink Modell gestoppt wird. Dazu wie folgt vorgehen:

Das File "arduino\_PWMOutput.p" muss das gleichnamige File in folgendem Ordner ersetzen: C:\ProgramData\MATLAB\SupportPackages\R2018a\toolbox\target\supportpackages\arduin obase\+codertarget\+arduinobase\+internal\<sup>1</sup>

Zu Sicherheit sollte davor das bereits existierende "arduino\_PWMOutput.p" file in "arduino\_PWMOutput\_old.p" umbenannt werden.

Nach erfolgreichem Ersetzen des "arduino\_PWMOutput.p" File, müssen folgende Befehle im Matlab command window ausgeführt werden.

- 1. clear pcode
- 2. clear classes
- 3. rehash toolboxcache

Nun wird der Widerstand nach beenden des Simulink Modells nicht weiter geheizt.

Nachdem alle Skripte auf dem Computer gespeichert, das Matlab File ausgetauscht und die nötigen Add.Ons installiert wurden, müssen der Ordner in dem die Skripte und Files auf dem PC gespeichert wurden noch zum Matlab Pfad hinzugefügt werden:

📣 MATLAB R2018a - acaden	nic use									- 🛛 🗙
HOME PLOTS	APPS						6 6 6 6 6	e 📮 🕐 🔻	Search Documentation	🔎 Sebastian 👻
New New New Op Script Live Script	Find Files	Save Workspace	New Vanable Open Variable • Clear Workspace • Favorites	Analyze Code Run and Time Clear Commands •	O Preferences     Layout Set Path A	So Com d-Ons Help ⊇ Lear	munity lest Support h MATLAB			-
4 + 17 1 + C +	Lisers + Sebastian +	CAR.	140L2	LUUE SHULKY	DAARWANGUT	HE BUILT	13			- 0
Current Folder	(	🕽 📝 Edito	or - C:\Users\Sebastian\Documents\I	ATLAB\Projektarbeit2\FregRes	ponse.m			(e) x	Workspace	۲
Name *     ARDUINO     Benutzerdefinierte C     Citavi 6	Office-Vorlagen	^ 1 - 2 - 3 -	<pre>if w == 0.0001 T_Filter = 0.1; elseif w == 0.01</pre>	ns_rr.m a Simulation_sv.m	A Simulation_sv1.m	U_USB_Mostelm	Freqkesponse.m A	-	Name *	Value
	essung_ert_rtw	4 - 5 - 7 - 8 - 9 - 10 -	<pre>jy = 1; ju = 1; T_filter = 5; elseif w &lt;= 0.07 jy = 0; ju = 0; T_filter = 5;</pre>							
Projekta     Projekta     Projekta     Projekta     Projekta     Projekta     Projekta     Projekta     Projekta	en w in Explorer ate Zip File	Engabe .	<pre>elseif w &lt;= 0.3 jy= 1; % j = 0 bedeut ju= 1; T_filter = 5; elseif w&gt;0.3</pre>	ed das letzte Peak wi	rd wird zur Berechn	ng verwendet, j	= 1 bedeuted das Vo	orletzt 🗧		
pzMove Ren     Sensors_ Delv	ame ete	F2 Loschen	nd Window					>		
Uebertra Nev	v Folder v File	>	in things							
FR_20.m Con FR_30.m Con	npare Selected Files/Folders npare Against	>								
FR_50.m. Cut	у	Strg+X Strg+C								
FR_70.m. Past	te	Stg+V		-						
FR_100.n Add	i to Path		Selected Folders							
Frequent Frequent Frequent Frequent	resh	F	Selected Folders and Subfolders	1						
Projektarbeit2 (Folder)	,	<b>,</b>							<	>
Ready										
E O Zur Suche	Text hier eingeben	Į,	) H   🗮 🙆 🌔	) 💿 🔯 🛽	🗉 🔟 🙂	🐴 🔕 🥠 🖡	r 🚯 🙆 r	~ ~ *	🕫 🖷 💩 🖉 🗔	08:40 14.03.2019 20

Dazu Matlab öffnen und in Current Folder Window den entsprechenden Ordner mit rechts anklicken  $\rightarrow$  Add to Path  $\rightarrow$  Select Folders and Subfolders.

<sup>&</sup>lt;sup>1</sup> Hinweis des Mathworks MATLAB Support Service

Anschließend können die verschiedenen Simulink Modelle "Projektarbeit2\_Steuerung", "Projektarbeit2\_SteuerungMitGW", "Projektarbeit2\_Regelung" und "Projektarbeit2\_DisturbanceCompensation" geöffnet und ausgeführt werden. Dazu einfach den Hardware Aufbau an dem Computer über die USB Ports verbinden, das jeweilige SIMULINK Modell öffnen und auf run klicken (siehe folgende Abbildung).



Bei erstmaliger Ausführung eines Modells kann es ein wenig dauern, bis der Code fertig kompiliert wurde (ca. 1 Minute).

Zur Überprüfung ob der Hardware Aufbau sowie die Softwareeinrichtung erfolgreich sind, soll nun das Simulink Modell "Projektarbeit2\_Steuerung.slx" geöffnet werden und durch Betätigung des "Run" Button gestartet werden. Das Programm endet entweder automatisch nach 2000 Sekunden oder durch manuelle Betätigung des "Stop" Buttons (rechts neben "Run"). Sobald das Programm beendet wird sollte ein Plot in der Form wie der unten dargestellte erscheinen:

