

OPGAVE 2 DIP-SWITCH STYRING AF TEKST PÅ OLED DISPLAY

1 Formål

Formålet med opgaven er at:

- at lære at skrive et struktureret C program der kan læse på input port
- at lære at ascii karakterer er binær encoded data
- at kunne benytte ascii alfabetet vist sidste side fra nr 32 til 127 decimalt (hexadecimal 0x10 til 7F) til at skrive en meningsfuld tekst ud på displayet styret af dip-switch og "clocket" in v.h.a. tryk-kontakt.

1.1 Læringsmål

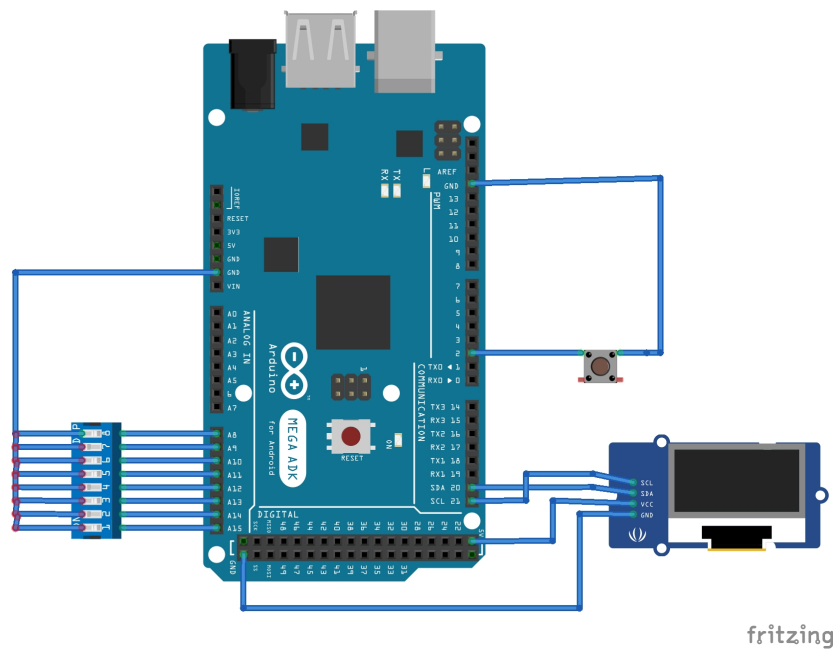
- At lære at programmere en input pind med intern pull-up modstand på ATmega2560
- At få benyttet og skrevet et program til at styre et OLED display til at skrive tekst ud på.
- At kunne beskrive hvilke registre en port består af og hvilke opsætninger af bit, der er nødvendige for input, pull up på indgang.

1.2 Schematic til forbindelse af OLED display og kontakt samt dip-switch

Dip switch forbindes til PORTK (A8-A15), Tryk-knap kontakt forbindes til PINE4 (D2) der skal programmeres/benyttes intern pull-up modstand til alle inputs. Se figur s. 2

2 Program

- Start et fælles journal dokument på f.eks. google doc. Og beskriv hvilke registre en port består af og hvilke opsætninger af bit der er nødvendige for input, output og pull up på indgang.
- Alle switches er aktive lave da der i programmet skal erklæres/anvendes interne pull-up modstande.
- Der skal fjernes pral på tryk-knap.
- Skriv nu et program i C der kan læse fra dip-switch en gang og kun en gang, for hvert tryk på kontakt.
- Læs minimum 3 forskellige binære værdier, der repræsenterer ascii karakterer ind fra dip switch i en array en for hvert tryk på kontakt.
- Og når alle værdierne er læst ind, så vises resultatet på OLED-displayet som f.eks. "hej", "alt", "Per", "123".
- Tryk switch skal være styret af eksternt interrupt (gennemgås lektion 3)



Figur 1: Forbindelser af komponenter til MEGA2560

3 Proces:

Benyt proces papiret på learn i programmerings process vist i bilag til opgaven

- Læs hele opgaven igennem
- Hvordan kunne en løsning til opgaven/delopgaverne se ud? Skitsere løsningen med pseudokode og/eller et flowchart for hver delopgave.
- Forestil dig en simuleret gennemkørsel af dit hypotetiske program/ dele af programmet. Benyt dit pseudo kode og flowchart.
- Hvordan kan C-kode for løsningen til delopgave/opgave se ud?
- Åben Microchip studio, start et nyt projekt, vælg GCC C executable, giv projektet et sigende navn og placere projektet et sted hvor du kan finde det igen. tryk ok.
- Udfører programmet det den hypotetiske gennemkørsel udførte under pkt. 3?

4 Fremgangsmåde:

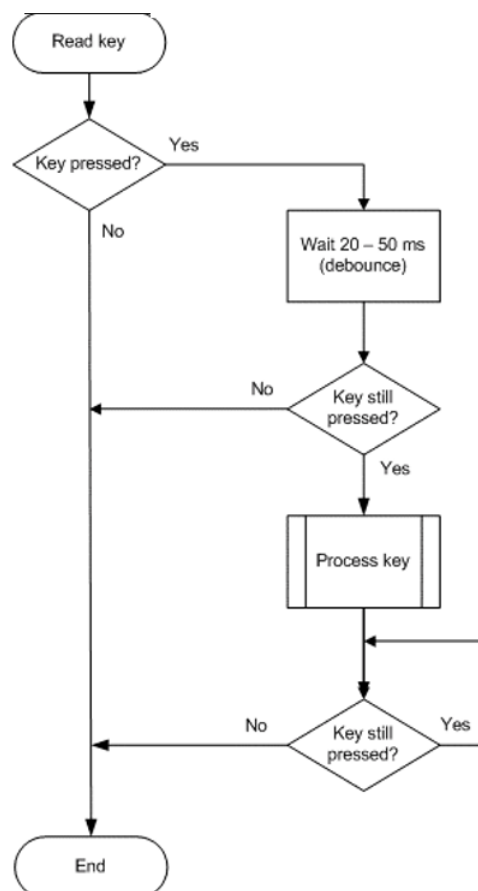
Når man skriver programmer skal man overveje at dele programmet op i passende funktioner og moduler (c fil .c og header fil .h).

- I main.c inde i while(1) udvikler man sin funktion ved at skrive hvad den skal gøre og teste delen af nede i microcontrolleren - når det virker
- Hvis jobbet i kodelinierne skal udføres mere end én gang så flyttes koden ud i en tilhørende funktion - kan være over eller under int main();

- (c) Kode der skal genanvendes i flere projekter som f.eks. fjerne pral flyttes så ud i en selvstændig c. fil med tilhørende header fil .h og funktionen gives en type og et sigende navn .eks. int debounce() {"koden fra main der udførte det ønskede"}
- (d) I header filen skrives så extern int debounce();

4.1 Debounce kontakt

- (a) Opret et nyt projekt i Atmel studio og indkluder OLED-display -c filer og header filer ind i projektet (ssd1306.h, ssd1306.c, i2c.h, i2c.c, data.h. Compiler og test at der kan skrives ud på display
- (b) Første del-problem er at kunne registrere at tryk-kontakt er trykket en og kun en gang - d.v.s pral skal fjernes. I int main() skrives nu et lille program der kan læse ind fra D2 (PINE4) i en variable og fjern pral ved at teste på valid tryk med delay mellem hver læsning på PINE4 (se flow chart) nedenfor. Definer en kontrol variabel og sæt den til 1 for sandt tryk og 0 for falsk.
- (c) For at få et delay mellem test på kontakt kan du benytte _delay_ms funktionen. Denne funktion er i et bibliotek #include <util/delay.h> i det er funktionen _delay_ms(int val) - giver et delay i "val milli sekunder". For at benytte det skal clock frekvensen F_CPU defineres før biblioteket inkluderes, som vist her #define F_CPU 16000000UL
#include <util/delay.h>



Figur 2: Flow char debounce

- (d) Upload program til MEGA2560-board og
- (e) Test at det virker i main(), f.eks. ved at erklære en tællevariabel, der tæller antal sande tryk og læs tæller-værdien ud på OLED-displayet for hvert tryk - juster evt. tiden hvis det ikke er robust. For at få vist tællerværdien korrekt, benyt sprintf funktionen fra afleveringsopgave 1, så tællerværdien konverteres til ascii og benyt funktionen sendStrXY() for at skrive det ud.
- (f) Når det virker flyt så koden der debouncer ud i en funktion med retur typen int - f.eks. placer den over int main() eller i sit eget c-modul med tilhørende header filer. Kald funktion fra while og test på returnværdi inde i main som ovenfor.
- (g) (Optional) Forbind et oscilloskop til input pin med kontakten, benyt single trigger mode og fang pral. Tag billede af det og sæt det ind i journal

4.2 Læs dip-switch

Næste delproblem er at læse dip-switch på PORTK (A8-A15) og benyt intern pull-up modstand

4.3 Læs en værdi fra switch ind i et array for hvert tryk på tryk-kontakt

Sæt nu overstående delprogrammer sammen, så der kan læses min. 3 forskellige værdier ind fra dip-switch i et array. Når der er 3 karakterer i array skal teksten skrives ud på displayet vha. sendStrXY funktionen som ord eller tal.

4.4 Eksternt interrupt på tryk kontakt - lektion 3

Som led i en iterativ udvikling så skal programmet optimeres, så der ikke benyttes kald til en debounce funktion, men i stedet skal benyttes eksternt interrupt til at fange tryk på kontakt.

5 Dokumentation og aflevering

Denne opgave er en gruppe aflevering og det skal ske senest tirsdag d. 22. feb. kl 23.59

Atmel Studio projekt zippes og lægges op sammen med en lille journal/rapport - der indeholder - en tekst formulering, der beskriver hvad opgaven går ud på, beskrivelse af hvilke registre en port består af og hvilke opsætninger af bit der er nødvendige for input, output og pull up på indgang. Beskriv hvad programmet skal kunne i en bullet liste, derefter et modul diagram over alle c-filer og header filer med tilhørende beskrivelser. Sæt jeres Flow diagrammer ind og forklar dem Og et test resultatet i form af en billede eller to der viser det virker. Eks modul-diagram er i mappen på Learn OLED display - datasheet og atmel studio project

God arbejdslyst

2022-02-02 Ole Schultz

6 Bilag Ascii alfabet

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040	 	Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	\$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS (backspace)	40	28	050	((72	48	110	H	H	104	68	150	h	h
9	9	011	TAB (horizontal tab)	41	29	051))	73	49	111	I	I	105	69	151	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC (escape)	59	3B	073	;	;	91	5B	133	[[123	7B	173	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174	|	
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135]]	125	7D	175	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177		DEL

Source: www.LookupTables.com

Figur 3: Tabellen viser omsætning fra decimal eller hex værdi til ascii karakter

7 bilag process

Programmeringsprocessen for datateknik og programmering

Bemærk det er en iterativ proces at programmere så 1 til 6 gennemløbes for hver delopgave.

1. Læs hele opgaven igennem?
 - (a) Giver opgaven mening? Diskuter opgaven med din gruppe. Hvis du er i tvivl - Spørg læreren eller hjælpelæreren
 - (b) Diskutere om opgaven kan deles op i mindre delopgaver.
2. Hvordan kunne en løsning til opgaven/delopgaverne se ud? Skitsere løsningen med pseudokode og/eller et flowchart for delopgaven.
 - (a) Hvilke fysiske enheder på microcontrolleren skal bruges?
 - (b) Hvor mange variabler skal benyttes? Hvilken type skal variabler have?
 - (c) Foretages der nogen udregninger? Hvordan se udregningen ud? Hvilket output forventes ved tilfældigt valgte inputs?
3. Forestil dig en simuleret gennemkørsel af dit hypotetiske program/ dele af programmet. Benyt dit pseudo kode og flowchart. Simuler at du giver kører hypotese programmet. Sker det forventede? Diskutere med din gruppemakker.
4. Hvordan kan C-kode for løsningen til delopgave/opgave se ud?
 - (a) Er behov for input for brugeren? Hvilke (tal, bogstaver, En sætning)?
 - (b) Hvilke register skal sættes? Hvilke bits skal sættes i registre? Hvad står der i databladet om de relevante registre? Husk at notere sidetallet i databladet for de benyttede registre.
 - (c) Skal der være et output? Hvilke(t)? Hvordan (OLED display, LED)?
 - (d) Diskutere med din gruppemakker.
 - (e) Har du noget tidligere kode som kan benytte til denne opgave? Er der et eksempel i bogen eller fra tidligere lektioner som kan benyttes?
5. Åben Microchip studio, start et nyt projekt, vælg GCC C executable, giv projektet et sigende navn og placere projektet et sted hvor du kan finde det igen. tryk ok. Vælg at der benyttes en ATmega2560
 - (a) Programmere den løsning i har fundet frem til for hver delopgave fundet i under 1.
 - (b) Compiler og Kør kode efter få linjers kodning, så man kan teste koden for fejl. Det gør det nemmere, at gennemskue hvor fejlen.
 - (c) Kommer der fejl så kan der debugges ved hjælp af simulatoren, OLED display eller URARTen
6. Udfører programmet det den hypotetiske gennemkørsel udførte under pkt. 3?
 - (a) Skal det justeres?
 - (b) Læs opgaven igennem igen og se om alle opgaver/delopgaver bliver udført
 - (c) Hvis der skal benyttes bruger input, Så få din gruppe kammerat til, at køre test programmet. Få gerne din gruppe kammerat til, at give feedback på din kode.