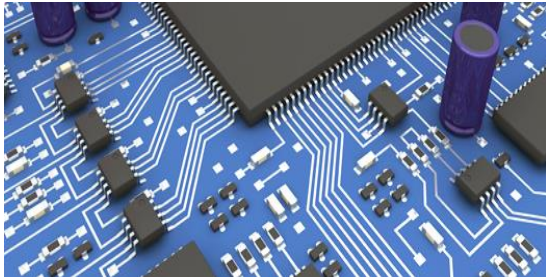


Aplikace Embedded systémů v Mechatronice



Michal Bastl
A2/713a

Aplikace Embedded systémů v Mechatronice

Obsah přednášky:

- Opakování
- LOOKUP tabulka
- Stavové automaty



Opakování

Co je PWM?

Proč se PWM používá?

Jak funguje generování PWM na MCU?

Co je třeba nastavit na PIC18 k PWM?

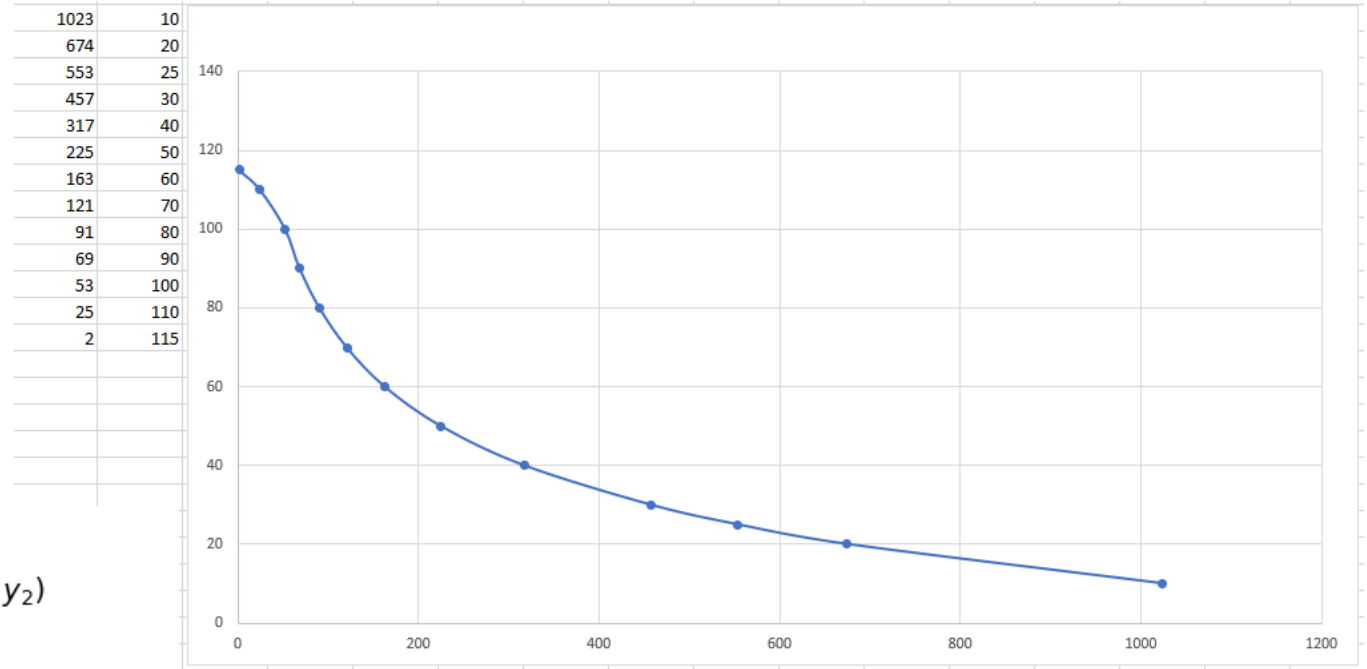
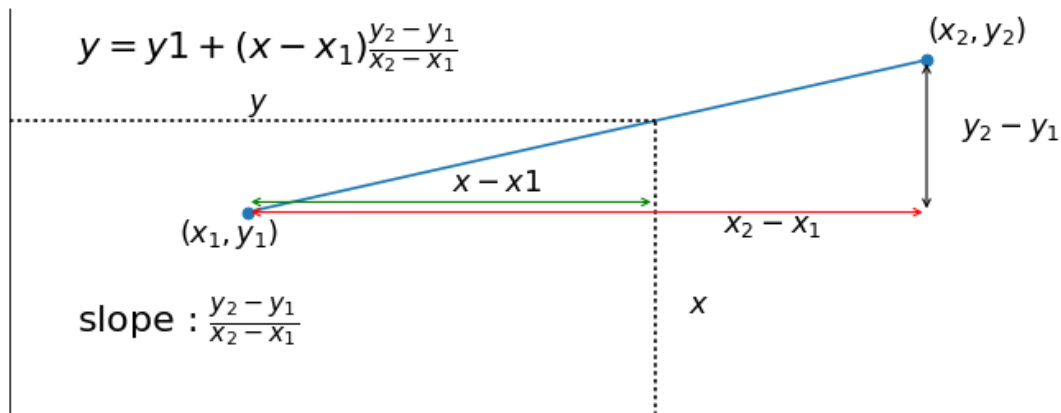
LOOKUP tabulka

- Je velmi používána technika k aproximaci, především nelineárních, funkcí.
- V podstatě se jedná o tabulku funkčních hodnot.
- Mezi hodnotami se nejčastěji provádí lineární interpolace.
- Příkladem může být měření pomocí nelineárního senzoru.
- Generování nelineárních funkcí sinus, tanh apod.

LOGARITHMS OF NUMBERS.													3
N	0	1	2	3	4	5	6	7	8	9	Differences.		
100	000000	0434	0808	1501	1734	2166	2508	2929	3401	3801	435	430 435 430	
101	4321	4751	5181	5609	6038	6466	6894	7321	7748	8174	1	44 45 46 47	
102	8600	9026	9451	9876	*300	*724	1147	1570	1993	2415	2	47 48 49 50	
103	012837	3259	3680	4100	4521	4940	5360	5779	6197	6616	3	174 175 176 177	
104	7033	7451	7868	8284	8700	9116	9532	9947	*361	*775	4	178 179 180 181	
105	021189	1603	2016	2428	2841	3252	3664	4075	4486	4896	5	218 219 220 221	
106	5306	5715	6125	6535	6942	7350	7757	8164	8571	8978	6	222 223 224 225	
107	9384	9789	*195	*600	1004	1408	1812	2216	2619	3021	7	226 227 228 229	
108	033424	3826	4227	4628	5029	5430	5830	6230	6629	7028	8	230 231 232 233	
109	7420	7825	8223	8620	9017	9414	9811	*207	*602	*998	9	234 235 236 237	
110	041893	1787	2182	2576	2969	3362	3755	4148	4540	4932	1	42 43 44 45	
111	5323	5714	6105	6495	6885	7275	7664	8053	8442	8830	2	125 126 127 128	
112	9218	9606	9993	*380	*766	1153	1538	1924	2309	2694	3	129 130 131 132	
113	053078	5403	5846	6289	6731	7173	7615	8056	8496	8936	4	133 134 135 136	
114	0905	7280	7696	8102	8508	8914	9320	9726	*370	*766	5	137 138 139 140	
115	000098	1075	1452	1829	2206	2582	2958	3333	3709	4085	6	141 142 143 144	
116	4458	4832	5206	5580	5953	6326	6699	7071	7443	7815	7	145 146 147 148	
117	8186	8557	8928	9298	9668	*338	*710	1082	1454	1826	8	149 150 151 152	
118	071892	2250	2617	2985	3352	3718	4085	4451	4816	5182	9	153 154 155 156	
119	5547	5912	6276	6640	7004	7368	7731	8094	8457	8819	1	157 158 159 160	
120	070181	9543	9904	*296	*696	*987	1347	1707	2067	2426	2	161 162 163 164	
121	082785	3144	3503	3861	4219	4576	4934	5291	5647	6004	3	165 166 167 168	
122	6360	6716	7071	7426	7781	8136	8490	8845	9198	9552	4	169 170 171 172	
123	9065	*258	*611	*963	1315	1667	2018	2370	2721	3071	5	173 174 175 176	
124	003422	3722	4122	4471	4820	5169	5518	5866	6215	6562	6	177 178 179 180	
125	0910	7267	7604	7951	8298	8644	8990	9335	9681	*20	7	181 182 183 184	
126	100371	9715	1059	1403	1747	2091	2434	2777	3119	3462	8	185 186 187 188	
127	3804	4146	4487	4828	5169	5510	5851	6191	6531	6871	9	189 190 191 192	
128	7210	7549	7888	8227	8565	8903	9241	9579	9916	*253	1	193 194 195 196	
129	110590	9926	1263	1599	1934	2270	2605	2940	3275	3609	2	197 198 199 200	
130	113043	4277	4611	4944	5278	5611	5943	6276	6608	6940	3	201 202 203 204	
131	7271	7603	7934	8265	8595	8926	9256	9586	9915	*245	4	205 206 207 208	
132	120574	9903	1231	1560	1888	2216	2544	2871	3198	3525	5	209 210 211 212	
133	3852	4178	4504	4830	5156	5481	5806	6131	6456	6781	6	213 214 215 216	
134	7165	7429	7753	8076	8399	8722	9045	9368	9690	*112	7	217 218 219 220	
135	130334	6655	*0077	1298	1619	1939	2260	2580	2900	3219	8	221 222 223 224	
136	3539	3858	4177	4496	4814	5133	5451	5769	6086	6403	9	225 226 227 228	
137	6721	7037	7354	7671	7987	8303	8618	8934	9249	9564	1	229 230 231 232	
138	9879	*194	*508	*822	1136	1450	1763	2076	2389	2702	2	233 234 235 236	
139	143015	3327	3639	3951	4263	4574	4885	5196	5507	5818	3	237 238 239 240	
140	146128	6438	6748	7058	7367	7676	7985	8294	8603	8911	4	241 242 243 244	
141	9219	9527	9835	*142	*449	*756	1063	1370	1676	1982	5	245 246 247 248	
142	152288	2504	2809	3105	3510	3815	4120	4424	4728	5032	6	249 250 251 252	
143	5530	5840	6143	6446	6749	7052	7355	7657	7959	8261	7	253 254 255 256	
144	8362	8664	8965	9266	9567	9868	*108	*469	*769	1068	8	257 258 259 260	
145	101268	1667	1967	2266	2564	2863	3161	3460	3758	4055	9	261 262 263 264	
146	4353	4650	4947	5244	5541	5838	6134	6430	6726	7022	1	265 266 267 268	
147	7317	7613	7908	8203	8497	8792	9086	9380	9674	9968	2	269 270 271 272	
148	170262	0555	0848	1141	1434	1726	2019	2311	2603	2895	3	273 274 275 276	
149	3180	3478	3769	4060	4351	4641	4932	5222	5512	5802	4	277 278 279 280	
150	176091	6381	6670	6959	7248	7536	7825	8113	8401	8689	5	281 282 283 284	
151	8077	9264	9552	9839	*126	*413	*699	*986	1272	1558	6	285 286 287 288	
152	181844	2129	2415	2700	2985	3270	3555	3839	4123	4407	7	289 290 291 292	
153	4091	4975	5259	5542	5825	6108	6391	6674	6956	7239	8	293 294 295 296	
154	7521	7803	8084	8366	8647	8928	9209	9490	9771	*51	9	297 298 299 300	
155	190532	0612	0892	1171	1451	1730	2010	2289	2567	2846	1	301 302 303 304	
156	3125	3403	3681	3959	4237	4514	4792	5069	5346	5623	2	305 306 307 308	
157	5000	0176	0453	0729	7005	7281	7556	7832	8107	8382	3	309 310 311 312	
158	8657	8932	9206	9481	9755	*20	*303	*577	*850	1124	4	313 314 315 316	
159	201307	1670	1943	2216	2488	2761	3033	3305	3577	3848	5	317 318 319 320	
N	0	1	2	3	4	5	6	7	8	9	Differences.		

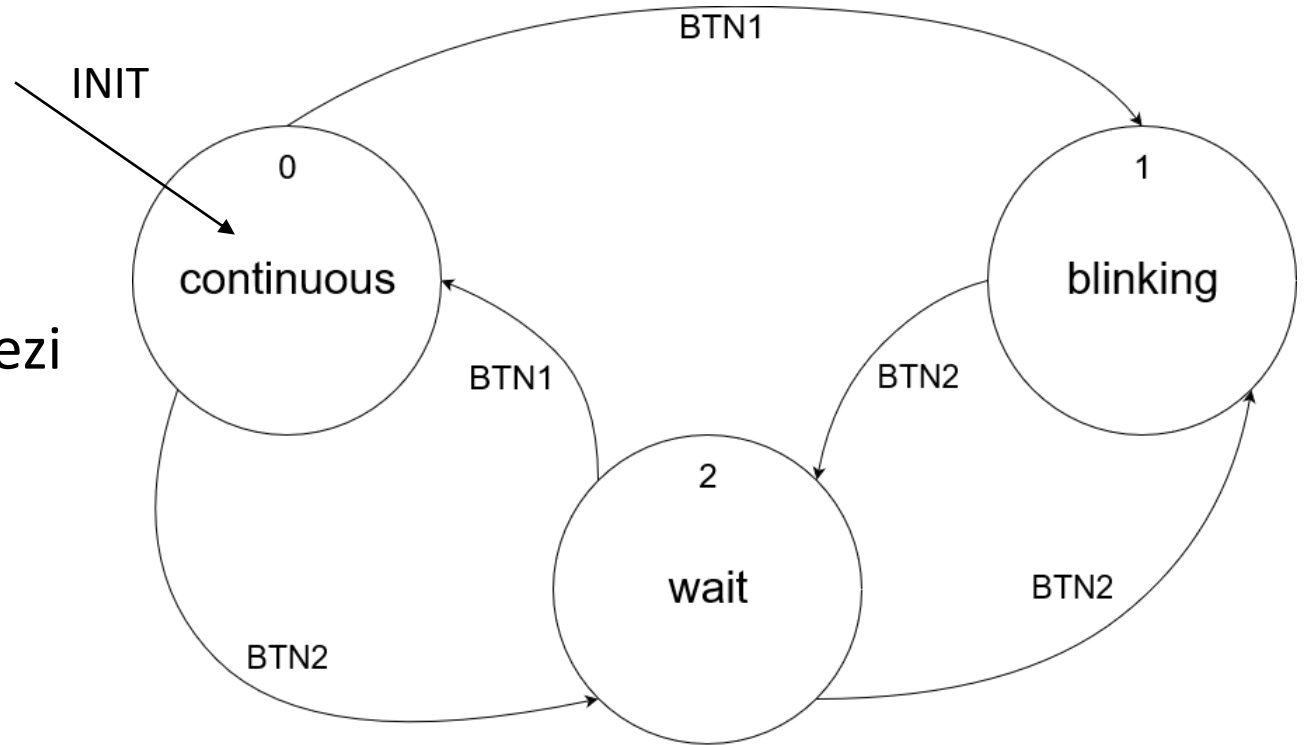
LOOKUP tabulka

- Měření teploty pomocí nelineárního senzoru NTC
- Nelineární měřicí charakteristiku nahradíme právě LOOKUP tabulkou
- osa x představuje hodnotu ADC převodníku
- osa y teplotu v °C



FSM

- FSM (konečný stavový automat)
- Je abstraktní stroj/automat, který je tvořen stavy.
- Je vhodný pro popis vývoje programu, který je řízen událostmi.
- Události mohou vyvolávat přechody mezi stavy.
- Výstup je definován aktuálním stavem.
- Koncept neříká nic o tom jak má být funkcionality implementována.



FSM

```
void FSM(void)
{
    static char STATE = 0;
    switch (STATE)
    {
        case 0:
            if (BTN1){
                __delay_ms(5);
                if(BTN1){
                    STATE = 1;
                }
            }
            if (BTN2){
                __delay_ms(5);
                if(BTN2){
                    STATE = 2;
                }
            }
            leds(0, 0);
            LCD_ShowString(1, "State:continuous");
            break;
    }
```

```
        case 1:
            if (BTN2){
                __delay_ms(5);
                if(BTN2){
                    STATE = 2;
                }
            }
            leds(1, 0);
            LCD_ShowString(1, "State: blinking ");
            break;
        case 2:
            if (BTN1){
                __delay_ms(5);
                if(BTN1){
                    STATE = 0;
                }
            }
            if (BTN2){
                __delay_ms(5);
                if(BTN2){
                    STATE = 1;
                }
            }
            leds(0, 1);
            LCD_ShowString(1, "State: wait ");
            break;
    }
}
```