

# Programátorská dokumentace SDS Big

**Obsah:**

Rozšiřující desky – sandwich boards.....	3
TopBoard LCD+KEY+DALI .....	3
Zapojení sandwich desky z hlediska uživatele.....	3
DALI master .....	3
Full C funkce pro komunikaci s deskou TopBoard LCD+KEY+DALI .....	4
Přehled.....	4
Funkce v detailu.....	4
Obsluha displaye.....	4
SDS_TOPB_LCDdraw.....	5
SDS_TOPB_LCDtext.....	6
SDS_TOPB_LCDbitmap.....	6
SDS_TOPB_LCDflip.....	7
API sběrnice DALI .....	7
SDS_TOPB_DALI_transmitCommand.....	7
SDS_TOPB_DALI_getReceivedResponse.....	9
API klávesnice.....	9
SDS_TOPB_keyboard_status .....	9
Vzorový program pro TopBoard LCD+KEY+DALI .....	10

## Rozšiřující desky – sandwich boards

- [LCD+KEY+DALI](#) - poskytuje grafické LCD plně ovladatelné z FULL-C programu, klávesnici s pěti tlačítky (stisk přenášen do FULL-C programu ke zpracování) a DALI master pro řízení světel a jiných DALI slave modulů
- LCD+KEY+CANCOR - poskytuje LCD, klávesnici, a rozhraní sběrnice CAN 2.0 A,B a rozhraní RS232 (COM6), vse přímo přístupné z FULL-C programu
- LCD+KEY+LORA - poskytuje LCD, klávesnici a připojení k LoRa WAN 868 síti (popř. FSK komunikaci přímo mezi více SDS moduly v rámci lokální sítě)
- SW-LORA - poskytuje pouze LoRa WAN 868 připojení
- SW-GSM - poskytuje pouze GSM připojení

## TopBoard LCD+KEY+DALI

TopBoard je doplňkem pro SDS Big. Jde o sandwich desku, která poskytuje uživatelské rozhraní k SDS Big, jako je grafický display a tlačítka. Dále tato deska poskytuje rozhraní ke sběrnici DALI. Touto deskou lze osadit i další moduly SDS, např. SDS-STSW, SDS-SMALL).

### Zapojení sandwich desky z hlediska uživatele

Vše na sandwich desce řídí kontrolér, který je na sandwich desce umístěn. Tento kontrolér má v sobě svůj program (firmware), který je připraven od výrobce a uživatel jej nemá důvod měnit.

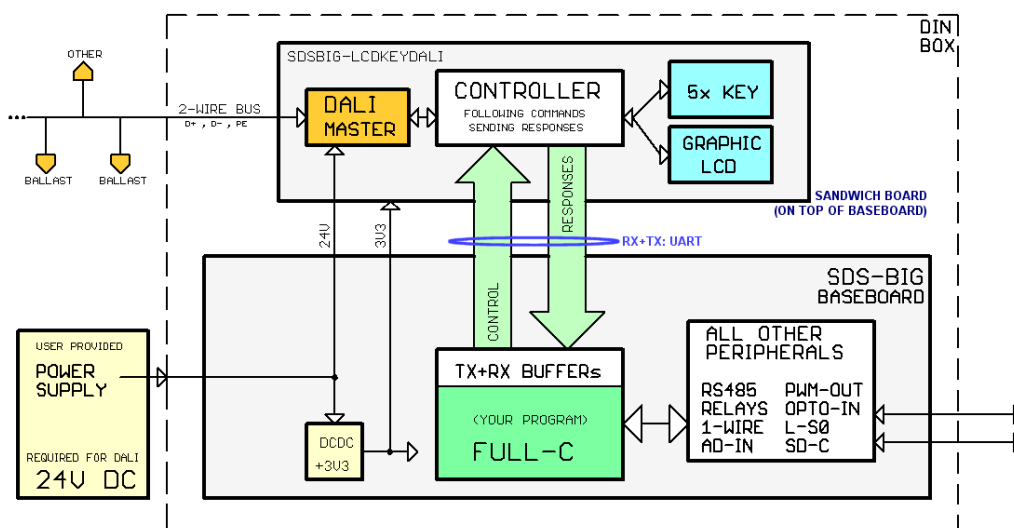
Kontrolér komunikuje se základní deskou (SDS-BIG) přes vyhrazené sériové rozhraní (SDS-BIG má několik samostatných sériových rozhraní pro různé účely).

Protokol, který je zde použit, je do detailu popsán a je určen pro uživatele, kteří si pomocí tohoto protokolu celou sandwich desku ovládají, ze svého FULL-C programu.

Pokud není v SDS-BIG nahrán uživatelský FULL-C program, jeho úlohu zastoupí SDS-BIG sám, a do sandwich desky posílá své vlastní údaje (takže na LCD se zobrazují hodnoty ze všech měřených vstupů).

Deska LCD-KEY-DALI poskytuje tyto funkce:

- Grafické LCD - výpis textu, grafika
- Tlačítka - pět samostatných tlačítek
- DALI master - viz popis dole



### DALI master

Sběrnice DALI je průmyslový standard pro řízení světel (zap/vyp, stmívání, scény, atd.).

Sběrnice má jednoho řídicího (master), což je SDS-BIG, a řadu podřízených (slave), což jsou světla, reléové moduly, moduly vstupů atd.

Nejčastěji je sběrnice DALI využita pro řízení a stmívání zářivkových svítidel (ballast), a také LED svítidel.

Délka kabelu sběrnice a počet podřízených na sběrnici je omezen především průřezem propojovacích vodičů v rámci rozvodu DALI signálu.

Potřebné jsou dva vodiče (pro DALI signál) a jeden společný PE (uzemnění - z bezpečnostních a ochranných důvodů).

DALI master může na sběrnici vysílat příkazy, a přijímat odpovědi od jednotlivých podřízených. Toto vše je k dispozici uživatelskému programu ve FULL-C, který posílá sadu příkazů do kontroleru v sandwich desce. Kontroler se pak postará o správné a přesné vyslání příkazů na DALI, a o přijetí odpovědí, které si pak FULL-C program může vyzvednout (nebo nemusí, u DALI to není nutné).

Chystáme i řadu vlastních modulů pro DALI (slave moduly), např. s relátky. Takto půjde snadno využít DALI sběrnici k distribuovanému řízení (např. ovládání světel v hale pomocí DALI světel, a současně i ovládání větrání haly pomocí DALI reléových modulů).

## Full C funkce pro komunikaci s deskou TopBoard LCD+KEY+DALI

*Upozornění:*

*Pro podporu funkcí této desky je vyžadován firmware SDS od verze 22.2.2016 nebo novější!*

### Přehled

Periferie	Signatura
Display	<code>void SDS_TOPB_LCDdraw(unsigned int X0, unsigned int Y0, unsigned int X1, unsigned int Y1, unsigned int DrawTYPE, unsigned int FLAGS);</code>
	<code>void SDS_TOPB_LCDtext(unsigned int X0, unsigned int ROW, unsigned int FLAGS, char *ptr, unsigned int len);</code>
	<code>void SDS_TOPB_LCDbitmap(unsigned int X0, unsigned int ROW, unsigned int FLAGS, char *ptr, unsigned int len);</code>
	<code>void SDS_TOPB_LCDflip(unsigned int FLAGS);</code>
DALI	<code>unsigned int SDS_TOPB_DALI_transmitCommand(unsigned int DALIaddress, unsigned int DALIdata);</code>
	<code>unsigned int SDS_TOPB_DALI_getReceivedResponse(unsigned int *ReceivedResponse);</code>
Klávesnice	<code>void SDS_TOPB_keyboard_status(unsigned int *changed, unsigned int *actual, unsigned int *intermediatePresses[6]);</code>

### Funkce v detailu

#### Obsluha displaye

TOPB nabízí grafický display, na který lze psát a kreslit.

Všechny tyto činnosti probíhají v bufferu na pozadí, a teprve na konkrétní příkaz (LCDflip) se tento buffer celý přenesou na display. Toto je důležité pro hladké zobrazení, bez blikání nebo poškozené animace - vždy novou scénu nakreslíte do paměťového bufferu, který představuje celý virtuální display, a ten se teprve na daný příkaz přenesou na skutečný display a zobrazí se.

**SDS\_TOPB\_LCDdraw****Účel**

Funkce vykreslí grafický objekt do bufferu.

**Signatura**

```
void SDS_TOPB_LCDdraw(unsigned int X0, unsigned int Y0, unsigned int X1, unsigned int Y1, unsigned int DrawType, unsigned int FLAGS);
```

**Argumenty**

Argument	Význam	Hodnoty																
unsigned int <b>X0, Y0, X1, Y1</b>	Grafické souřadnice objektu	X: 0 .. 131 Y: 0 ..31																
Unsigned int <b>DrawType</b>	typ grafického objektu, který se má vykreslit	<ul style="list-style-type: none"> <li>- <b>0: PIXEL</b>: vykreslí bod na souřadnici X0 / X0. X1,Y1 jsou ignorovány.</li> <li>- <b>32: LINE</b>: vykreslí jeden pixel širokou čáru od X0,Y0 do X1,Y1</li> <li>- <b>64: ELLIPSE</b>: vykreslí elipsu od X0,Y0 do X1,Y1</li> <li>- <b>96: RECTANGLE</b>: vykreslí obdélník od X0,Y0 do X1,Y1</li> <li>- <b>224: FLIP</b>: stejný význam jako funkce SDS_TOPB_LCDflip</li> </ul>																
unsigned int <b>FLAGS</b>	Parametr <i>FLAGS</i> je bitová kombinace, určující způsob vykreslení pixelů objektu, která navíc svým významem závisí na hodnotě <i>DrawType</i>	Bity MSB to LSB: <table border="1" style="margin: 10px auto;"> <tr> <td style="text-align: center;">7</td> <td style="text-align: center;">6</td> <td style="text-align: center;">5</td> <td style="text-align: center;">4</td> <td style="text-align: center;">3</td> <td style="text-align: center;">2</td> <td style="text-align: center;">1</td> <td style="text-align: center;">0</td> </tr> <tr> <td colspan="3" style="text-align: center;">RSVD</td> <td style="text-align: center;">FLIP</td> <td colspan="2" style="text-align: center;">SCI</td> <td style="text-align: center;">FILLED</td> <td style="text-align: center;">ADDITIONAL</td> </tr> </table> <ul style="list-style-type: none"> <li>- <b>RSVD</b>: nastavte na nulu.</li> <li>- <b>FLIP</b>: nastavte na 1, pokud se má buffer ihned vykreslit</li> <li>- <b>SCI</b>: určí způsob vykreslení bodů objektu               <ul style="list-style-type: none"> <li>SCI == 00 SET PIXEL (černá barva)</li> <li>SCI == 01 CLEAR PIXEL (bílá / podklad)</li> <li>SCI == 11 INVERT PIXEL (obráť hodnotu pixelu)</li> </ul> </li> <li>- <b>FILLED</b>: použito pouze pro <i>DrawType</i> 64 a 96 (ELLIPSE a RECTANGLE). Pokud je tento příznak nastaven, provede se vyplnění vnitřní plochy objektu podle SCI.</li> <li>- <b>ADDITIONAL</b>: použito pouze pro <i>DrawType</i> 224 (FLIP). Pokud je tento příznak nastaven, je buffer vyčištěn a provede se FLIP.</li> </ul>	7	6	5	4	3	2	1	0	RSVD			FLIP	SCI		FILLED	ADDITIONAL
7	6	5	4	3	2	1	0											
RSVD			FLIP	SCI		FILLED	ADDITIONAL											

**POZOR:** starší firmware v TOPB desce ne vždy umí všechny možné grafické prvky - zašlete výrobek výrobci pro aktualizaci firmware.

**SDS\_TOPB\_LCDtext****Účel**

Vykreslí text do bufferu.

**Signatura**

```
void SDS_TOPB_LCDtext(unsigned int X0, unsigned int ROW, unsigned int FLAGS, char *ptr, unsigned int len);
```

**Argumenty**

Argument	Význam	Hodnoty																
unsigned int <b>X0</b>	určuje posun o daný počet pixelů na řádku	0 .. 131																
Unsigned int <b>ROW</b>	vybírání řádek (LCD má čtyři řádky)	0 .. 3																
unsigned int <b>FLAGS</b>	Parametr <i>FLAGS</i> je bitová kombinace určující způsob vykreslení jednotlivých pixelů textu.	Bity MSB to LSB: <table border="1" style="margin-left: 20px;"> <tr> <td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td> </tr> <tr> <td colspan="2">RSVD</td><td>FLIP</td><td colspan="2">SCI</td><td colspan="3">RSVD</td> </tr> </table> <ul style="list-style-type: none"> <li>- <b>RSVD</b>: nastavte na nulu.</li> <li>- <b>FLIP</b>: nastavte na 1, pokud se má buffer ihned vykreslit</li> <li>- <b>SCI</b>: určuje způsob vykreslení bodů textu               <ul style="list-style-type: none"> <li>SCI == 00 POSITIVE Text se vykreslí černě</li> <li>SCI == 01 NEGATIVE Text se vykreslí bíle na černém podkladu</li> <li>SCI == 11 INVERT TEXT</li> </ul> </li> </ul>	7	6	5	4	3	2	1	0	RSVD		FLIP	SCI		RSVD		
7	6	5	4	3	2	1	0											
RSVD		FLIP	SCI		RSVD													
Char *ptr	Ukazatel na text, který se má zobrazit																	
Unsigned int len	Počet znaků v textu	0...22																

**SDS\_TOPB\_LCDbitmap****Účel**

Přenos malé bitmapy (blok pixelů) do bufferu - tato funkce umí přenést bitmapu na jeden určený řádek. Pro pokrytí celého displeje, který má čtyři řádky, je tedy potřeba použít tuto funkci čtyřikrát, vždy se správnými parametry.

**Signatura**

```
void SDS_TOPB_LCDbitmap(unsigned int X0, unsigned int ROW, unsigned int FLAGS, char *ptr, unsigned int len);
```

**Argumenty**

Argument	Význam	Hodnoty																
unsigned int <b>X0</b>	určuje posun o daný počet pixelů na řádku	0 .. 131																
Unsigned int <b>ROW</b>	vybírání řádek (LCD má čtyři řádky)	0 .. 3																
unsigned int <b>FLAGS</b>	Parametr <i>FLAGS</i> je bitová kombinace určující způsob vykreslení jednotlivých pixelů textu.	Bity MSB to LSB: <table border="1" style="margin-left: 20px;"> <tr> <td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td> </tr> <tr> <td colspan="2">RSVD</td><td>FLIP</td><td colspan="2">RSVD</td><td colspan="3">ADDITIONAL</td> </tr> </table> <ul style="list-style-type: none"> <li>- <b>RSVD</b>: nastavte na nulu.</li> <li>- <b>FLIP</b>: nastavte na 1, pokud se má buffer ihned vykreslit</li> <li>- <b>ADDITIONAL</b>: pokud je tento bit nastaven, je celý buffer vyplněn výchozím obrázkem loga.</li> </ul>	7	6	5	4	3	2	1	0	RSVD		FLIP	RSVD		ADDITIONAL		
7	6	5	4	3	2	1	0											
RSVD		FLIP	RSVD		ADDITIONAL													
Char *ptr	Ukazatel na paměť s bitmapou																	
Unsigned int len	Velikost obrazových dat																	

**SDS\_TOPB\_LCDflip****Účel**

Základní funkce pro překreslení (FLIP) paměťového bufferu (virtuálního LCD) na skutečný (hardware) LCD. Principem je provádění veškerých nových kreseb na pozadí, a až nakonec, najednou, se tyto přenesou na skutečný display.

**Signatura**

```
void SDS_TOPB_LCDflip(unsigned int FLAGS);
```

**Argumenty**

Argument	Význam	Hodnoty
Unsigned int <b>FLAGS</b>	S výhodou můžete použít <code>FLAGS==1</code> pro okamžité smazání celého displaye. Běžná funkce má <code>FLAGS==0</code> , kdy dojde jen k přenesení celého bufferu na display.	0: FLIP: pouze přenesou obsah z bufferu na display 1: CLEAR and FLIP: vyčistí buffer a pak jej přenesou na displej.

**API sběrnice DALI**

Vše je popsáno do detailu v příslušné normě.

Na sběrnici je jeden MASTER (zde SDS) a množství SLAVE jednotek - vždy záleží na konkrétním projektu.

Každá SLAVE jednotka má tyto tři určené adresy:

- BROADCAST - globální konstantní adresa (0xFF) na které slyší naprosto všechny jednotky najednou
- GROUP - skupinová adresa, tuto adresu má jedna nebo více jednotek společně, nastavuje se na určenou hodnotu podle projektu
- DIRECT - krátká adresa, tuto adresu má vždy jen jedna jediná jednotka, nastavuje se na určenou hodnotu podle projektu

Každé SLAVE jednotce na sběrnici DALI se tedy nastaví tyto tři hodnoty (0xFF a dvě volitelné). Jaké to jsou, si samozřejmě musíte určit vy sami - protože jen vy víte, jak a co máte zapojeno a k čemu to chcete použít. Skupinové adresy jsou výhodné, protože se dá ovládat více prvků jedním příkazem (např. hromadně zhasnout skupina světel) najednou = bez nutnosti postupného posílání adresných příkazů každé jednotce, což pak není vizuálně pěkné (světla v tomto příkladu by zhasnala postupně).

Jak se adresy nastavují - záleží VŽDY na výrobci dané SLAVE jednotky. Někdy je prostřednictvím jumperů / DIP spínačů / atd. na výrobku, a někdy se to dělá tak, že se jednotka připojí k osobnímu počítači a přes určený program se do ní zapíše tyto adresy (skupinová a přímá adresa).

Pokud se nějaká už instalovaná jednotka pokazí a je potřeba ji nahradit novou (výměna) pak je potřeba tuto novou jednotku správně naadresovat - nastavit jí stejné adresy jaké měla ta původní kterou chcete vyměnit.

**Způsob komunikace**

MASTER (SDS) umí poslat příkaz do SLAVE jednotek na sběrnici, a současně umí i přijmout odpověď (pokud je očekávána odpověď od SLAVE jednotky).

Pokud je požadována odpověď (např. na dotaz na nastavenou hodnotu nebo stav tlačítka), pak je potřeba na MASTERu použít příkazy které adresují SLAVE jednotky pouze přes přímou (DIRECT) adresu!

**SDS\_TOPB\_DALI\_transmitCommand****Účel**

SDS poskytuje přímý přístup na sběrnici. Funkce umožňuje odeslat DALI příkaz (vždy 2 bajty, adresa+data), z SDS v roli DALI MASTER. Funkce vrací 1 byte odpovědi.

**Signatura**

```
unsigned int SDS_TOPB_DALI_transmitCommand(unsigned int DALIaddress, unsigned int DALIdata);
```

### Argumenty a návratová hodnota

Argument	Význam	Hodnoty																										
unsigned int <b>DALIaddress</b>	Adresa DALI	Byte 0 .. 255  <b>Struktura:</b> YAAA AAAS  <b>Y:</b> typ adresy 0: krátká adresa (DALI: short address) 1: skupinová adresa (DALI: group address or collective call)  <b>A:</b> bity hodnoty adresy  <b>S:</b> definuje význam datového byte:  0: hodnota přenesená v data byte bude použita pro přímou kontrolu jasu svítidla 1: hodnota v data byte obsahuje příkaz DALI.																										
Unsigned int <b>DALIdata</b>	DALI data byte	<b>Stručný přehled vybraných DALI příkazů:</b> <table border="1"> <thead> <tr> <th>Příkaz</th> <th>Význam</th> </tr> </thead> <tbody> <tr> <td>0x00</td> <td>Zhasne svítidlo</td> </tr> <tr> <td>0x01</td> <td>Plynule zvýší úroveň jasu svítidla</td> </tr> <tr> <td>0x02</td> <td>Plynule sníží úroveň jasu svítidla</td> </tr> <tr> <td>0x03</td> <td>Zvýší úroveň jasu svítidla</td> </tr> <tr> <td>0x04</td> <td>Sníží úroveň jasu svítidla</td> </tr> <tr> <td>0x05</td> <td>Nastaví jas na minimum</td> </tr> <tr> <td>0x06</td> <td>Nastaví jas na maximum</td> </tr> <tr> <td>0x20</td> <td>Reset svítidla</td> </tr> <tr> <td>0x90</td> <td>Vrátí status svítidla</td> </tr> <tr> <td>0xA0</td> <td>Vrátí aktuální úroveň jasu svítidla</td> </tr> <tr> <td>0xA1</td> <td>Vrátí maximální povolenou úroveň jasu svítidla</td> </tr> <tr> <td>0xA2</td> <td>Vrátí minimální povolenou úroveň jasu svítidla</td> </tr> </tbody> </table> <i>Pro kompletní seznam příkazů nebo pro příkazy, které jsou specifické pro konkrétní zařízení doporučujeme normu DALI nebo dokumentaci k danému zařízení.</i>	Příkaz	Význam	0x00	Zhasne svítidlo	0x01	Plynule zvýší úroveň jasu svítidla	0x02	Plynule sníží úroveň jasu svítidla	0x03	Zvýší úroveň jasu svítidla	0x04	Sníží úroveň jasu svítidla	0x05	Nastaví jas na minimum	0x06	Nastaví jas na maximum	0x20	Reset svítidla	0x90	Vrátí status svítidla	0xA0	Vrátí aktuální úroveň jasu svítidla	0xA1	Vrátí maximální povolenou úroveň jasu svítidla	0xA2	Vrátí minimální povolenou úroveň jasu svítidla
Příkaz	Význam																											
0x00	Zhasne svítidlo																											
0x01	Plynule zvýší úroveň jasu svítidla																											
0x02	Plynule sníží úroveň jasu svítidla																											
0x03	Zvýší úroveň jasu svítidla																											
0x04	Sníží úroveň jasu svítidla																											
0x05	Nastaví jas na minimum																											
0x06	Nastaví jas na maximum																											
0x20	Reset svítidla																											
0x90	Vrátí status svítidla																											
0xA0	Vrátí aktuální úroveň jasu svítidla																											
0xA1	Vrátí maximální povolenou úroveň jasu svítidla																											
0xA2	Vrátí minimální povolenou úroveň jasu svítidla																											
Návratová hodnota (unsigned int)	Výsledek odeslání příkazu	0: odeslání selhalo 1: odeslání ok																										

### Opakování příkazu

Některé DALI příkazy vyžadují podle specifikace normy DALI opakování příkazu nejpozději do 100 ms. U následujících příkazů to podle této normy dělá SDS za vás (tzn. FULL-C program zavolá funkci pro odeslání příkazu jednou, a SDS to zopakuje dvakrát, tzn. automaticky odešle 2x na sběrnici).

Automatizované opakování v rámci SDS je zde z důvodů že norma DALI požaduje opakování s maximálním prodloužením pod 100 milisekund - což typicky přes FULL-C zajistit nelze.

Automaticky opakované příkazy při odesílání z SDS MASTER do sběrnice DALI:

- Pro ADDRESS==0x01 a současně DATA v hodnotě z rozsahu 0x20 (včetně) do 0x80 (včetně)
- Pro ADDRESS==0xA5 (Initialize CMD) a libovolné DATA
- Pro ADDRESS==0xA7 (Randomize CMD) ale jen když DATA==0x00

### Doplňující informace

Pokud je na sběrnici DALI porucha, všechny slave jednotky typicky přecházejí do "neřízeného" režimu. To znamená, že se přestanou chovat podle toho jak byly přes DALI nastaveny, a typicky přejdou do "100% zapnuto" režimu. Toto je v pořádku a výrobci slave jednotek to tak chtějí (tím pádem se slave jednotky chovají jen jako "hloupý napáječ světla" který se dá ovládat vypínačem).



Pozn. před spuštěním DALI komunikace vyčkejte ve svém programu (ideálně) několik vteřin, zejména je-li délka vedení DALI sběrnice velká (dlouhé kabely). Chvilu trvá než se celý systém "nabije" a dokud se to nestane, nelze přenášet DALI příkazy.

### SDS\_TOPB\_DALI\_getReceivedResponse

#### Účel

Funkce vrátí výsledek zaslání příkazu a příznakové informace o volání a zpracování odpovědi.

#### Signatura

```
unsigned int SDS_TOPB_DALI_getReceivedResponse(unsigned int *ReceivedResponse);
```

#### Argumenty a návratová hodnota

Argument	Význam	Hodnoty										
unsigned int <b>*ReceivedResponse</b>	Na adresu tohoto ukazatele se uloží data (payload) přijatá na sběrnici, pokud jsou platná	0 .. 255										
<b>Návratová hodnota</b> unsigned int	Stav úspěchu komunikace RXFlags	Příznaky MSB - LSB <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>31..</th> <th>3</th> <th>2</th> <th>1</th> <th>0</th> </tr> </thead> <tbody> <tr> <td>Rezervováno</td> <td>INVDATA</td> <td>REPEATED</td> <td>FAILURE</td> <td>GOT-RX</td> </tr> </tbody> </table> INVADATA      označuje příjem neplatných dat (chyba slave, dva nebo více slave vysílají odpověď najednou - viz příkaz COMPARE, apod. REPEATED      informuje že došlo k automatickému opakování vysílání dle požadavku normy DALI FAILURE        označuje chybový stav sběrnice (zkrat, nadproud, chyba HW vysílače). GOT-RX         Nejdůležitější je flag "GOT-RX". Pokud je tento bit nastaven na 1, pak jsou data v "DALI_RX" platná a mohou se použít.	31..	3	2	1	0	Rezervováno	INVDATA	REPEATED	FAILURE	GOT-RX
31..	3	2	1	0								
Rezervováno	INVDATA	REPEATED	FAILURE	GOT-RX								

### API klávesnice

#### SDS\_TOPB\_keyboard\_status

#### Účel

TOPB nabízí pět kláves (tlačítek), které jsou interně ošetřeny proti zákmitům (nemusíte se dále ve FULL-C programu o toto starat).

Každý stisk je zaznamenán, takže k dispozici pro FULL-C program je nejen aktuální stav každého tlačítka (stisknuto/uvolněno), ale i počet stisknutí mezi jednotlivými dotazy programu (takže když uživatel rychle mačká tlačítko, tak FULL-C program vždy ví, kolikrát to bylo).

Tato funkce je jedinou funkcí ve FULL-C, která zajišťuje kompletní obsluhu zjišťování stavu klávesnici. Volat tuto funkci je potřeba jen tak často, jak moc je potřeba mít nízkou odezvu.

#### Signatura

```
void SDS_TOPB_keyboard_status(unsigned int *changed, unsigned int *actual, unsigned int *intermediatePresses[6]);
```

#### Argumenty

Argument	Význam	Hodnoty												
unsigned int <b>*changed</b>	Hodnota <i>changed</i> ukazuje, zda-li se od posledního zavolání funkce změnil stav některého z tlačítek.	Pole pěti bitů, bit je nastaven v případě změny odpovídajícího tlačítka  Bity MSB to LSB: <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>Bit</th> <th>4</th> <th>3</th> <th>2</th> <th>1</th> <th>0</th> </tr> </thead> <tbody> <tr> <td>tlačítko</td> <td>5</td> <td>4</td> <td>3</td> <td>2</td> <td>1</td> </tr> </tbody> </table>	Bit	4	3	2	1	0	tlačítko	5	4	3	2	1
Bit	4	3	2	1	0									
tlačítko	5	4	3	2	1									
Unsigned int <b>*actual</b>	Hodnota <i>actual</i> obsahuje aktuální stavy všech tlačítek	Bitově kódovaný stav tlačítek  Bity MSB to LSB: <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>Bit</th> <th>4</th> <th>3</th> <th>2</th> <th>1</th> <th>0</th> </tr> </thead> <tbody> <tr> <td>tlačítko</td> <td>5</td> <td>4</td> <td>3</td> <td>2</td> <td>1</td> </tr> </tbody> </table>	Bit	4	3	2	1	0	tlačítko	5	4	3	2	1
Bit	4	3	2	1	0									
tlačítko	5	4	3	2	1									

unsigned int <b>*<i>intermediatePresses</i>[6]</b>	Prakticky všechny funkce lze řešit přes pole <i>intermediatePresses</i> [6]. Pro každé tlačítko v tomto poli je udán počet stisků které uživatel provedl, od posledního čtení této hodnoty prostřednictvím volání této funkce. Tzn. po každém zavolání funkce se tyto vnitřní počítadla nulují.	Počet stisknutí. Validní jsou indexy 1 až 6. Fyzické rozložení tlačítek a jejich index v poli <i>intermediatePresses</i> : <table border="1" data-bbox="970 293 1305 387" style="margin: 10px auto;"> <tr> <td style="background-color: black; color: white; text-align: center;">1</td> <td style="background-color: black; color: white; text-align: center;">3</td> <td style="background-color: black; color: white; text-align: center;">5</td> </tr> <tr> <td style="background-color: black; color: white; text-align: center;">2</td> <td style="background-color: gray; text-align: center;"> </td> <td style="background-color: black; color: white; text-align: center;">4</td> </tr> </table>	1	3	5	2		4
1	3	5						
2		4						

## Vzorový program pro TopBoard LCD+KEY+DALI

Tento program je funkčním příkladem pro demonstraci všech funkcí rozšiřující desky.

Najdete v něm použity všechny funkce API TopBoardu LCD+KEY+DALI v kontextu smysluplného programu.

Co program dělá?

- po startu spustí jednoduché grafické demo – funkce LCDGraficDemo(), ze kterého je možné pochopit, jak funguje práce s bitovými mapami. Toto demo trvá několik vteřin, potřebných současně pro uklidnění DALI sběrnice po zapnutí zařízení, takže následně spuštěná DALI komunikace by měla probíhat bez potíží.
- Po ukončení grafického dema se spouští funkce daliDemo(). Tato funkce prohledá vyhledá připojená DALI zařízení na sběrnici a umožní obsluhu ovládat jasy prvních devíti nalezených zařízení.

Program si stáhněte z naší wiki.

*Pokud při běhu programu vypíše SDS do konzoly out of memory, zkompilujte program s větším zásobníkem (Manual STACK size selection).*