Arduino - ez_SIPO8_lib Crib Sheet

Library Class Initiation {download a copy of the full User Guide here from github}

Class Name: **SIPO8**

Class Initiation Syntax: **SIPO8** my_SIPOs(Num_SIPOs, Num_timers);

where 'my_SIPOs' is <u>any</u> name you wish to use for your project and 'Num_SIPOs' and 'Num_timers' are the number of SIPO ICs and timers you wish to define. For example:

1.
```
SIPO8 my_SIPOs( 1, 0); // define 1 SIPO, 0 timers
```

2.
```
#define Max_SIPOs    8
#define Max_timers   4
SIPO8 my_SIPOs(Max_SIPOs, Max_timers);
```

3.
```
etc
```

declare the class instance early in your code, for example after any SIPO data but prior to the setup() function

Available User Accessible Library Macros Definitions

Macro Name – #define	Value	Associated Functions
pins_per_SIPO	8	The number of bits/output pins of a virtual/physical SIPO and on which the core design of the library is based.
create_bank_failure	-1	The return value from the create_bank function, if a request to create a new bank for the specified number of SIPOs cannot be met, e.g. too few remaining unallocated SIPOs in the unused pool.
pin_read_failure	-1	The return value from the read_array_pin & read_bank_pin functions, if a specified pin is not an active pin.
pin_invert_failure	-1	The return value from the invert_array_pin & invert_bank_pin functions, if a specified pin is not an active pin.
pin_set_failure	-1	The return value from the set_array_pin & set_bank_pin functions, if a specified pin is not an active pin.
bank_not_found	-1	The return value from the get_bank_from_pin, set_bank_SIPO, invert_bank_SIPO &

Ron Bentley, Stafford UK

Macro Name – #define	Value	Associated Functions
		`read_bank_SIPO` functions, if a specified pin/SIPO does not occupy a defined SIPO bank.
`SIPO_not_found`	−2	The return value from the `set_bank_SIPO`, `invert_bank_SIPO` & `read_bank_SIPO` functions, if a specified SIPO not in range for given bank.
`timer0`	0	Eight predefined macros are provided that may be used for 1 – 8 configured timers. Beyond eight, the end user will either reference timers explicitly within the limit of the number of timers set up at class initiation , e.g. 9, 10, etc, or by using his/her own declared macros for such purposes.
`timer1`	1	
`timer2`	2	
`timer3`	3	
`timer4`	4	
`timer5`	5	
`timer6`	6	
`timer7`	7	
`elapsed`	true	Return value of the `SIPO8_timer_elapsed` function if the defined elapse period has completed for the specified timer.
`not_elapsed`	!elapsed	Return value of the `SIPO8_timer_elapsed` function if the defined elapse period for the specified timer has not yet completed.
`active`	true	Used internally by the SIPO8 timer functions to judge if the specified timer is active.
`not_active`	!active	Used internally by the SIPO8 timer functions to judge if the specified timer is inactive.

Available User Accessible Library Variables

SIPO8 Control Sruct(ure)	Purpose
`struct SIPO_control {`	For each bank created (`create_bank` function) maintains a record of:
`uint8_t bank_data_pin;`	Digital pin allocated to the data input line to a SIPO bank
`uint8_t bank_clock_pin;`	Digital pin allocated to the clock input line to a SIPO bank
`uint8_t bank_latch_pin;`	Digital pin allocated to the latch input line to a SIPO bank
`uint8_t bank_num_SIPOs;`	The number of 8bit SIPO ICs grouped under a bank
`uint16_t bank_low_pin;`	The first output pin number for the bank (absolute pin address)
`uint16_t bank_high_pin;`	The last output pin number for the bank (absolute pin address)
`} * SIPO_banks;`	`SIPO_banks` is the active/working data structure used throughout the library for managing and controlling SIPO data. It is available for direct user code inspection.

SIPO8 Timer Sruct(ure)	Purpose
`struct timer_control {`	For each timer created maintains a record of:
` bool timer_status;`	Current status of a timer - `active` or `not_active`
` uint32_t start_time;`	The millis time when a timer is started
`} * timers;`	`timers` is the active/working data structure used throughout the library for managing and controlling timers. It is available for direct user code inspection.

Pin Status Bytes (PSB)	Purpose/definition
`uint8_t * pin_status_bytes;`	An array sized and created to be the number of 8 bit unsigned bytes required to map the maximum number of SIPO ICs defined at class initiation. That is, one status byte per required 8bit SIPO IC. This structure is the virtual SIPO output pin map and represents the entire SIPO output pin array pool. The `pin_status_bytes` array is used to record the status (`HIGH` or `LOW`) of each physical SIPO output pin. It is available for direct user code inspection.

Other User Accessible Variables/Declarations	Purpose/definition
`uint8_t  max_SIPOs`	The maximum number of SIPO ICs defined at class initiation by the user code.
`uint16_t max_pins`	The maximum number of SIPO output pins defined for the SIPO pool. Note that this is not the number of active SIPO output pins, but is the maximum number of SIPO output pins available for allocation to SIPO banks via calls to the `create_bank` function. Therefore, `max_pins` `>=` `num_active_pins`.
`uint16_t num_active_pins`	The number of <u>active</u> SIPO output pins allocated from the SIPO output pin pool. Allocated by the `create_bank` function – see above.
`uint8_t num_pin_status_bytes`	The number of 8 bit unsigned bytes allocated to map the maximum number of SIPO output pins. For example, if the max number of SIPO output pins is 128, then this value will be 128/8 = 16 bytes.
`uint8_t  num_banks`	The total number of SIPO banks created by using `create_bank` function.
`uint8_t  bank_SIPO_count`	The total number of SIPO ICs in use by defined SIPO banks. Note that `bank_SIPO_count` `<=` `max_SIPOs`.
`uint8_t  max_timers`	The number of timers defined at time of class initiation by the user code.

Ron Bentley, Stafford UK

Available User Accessible Library Functions

Function/Method	Parameters	Returned Value(s)	Comments
SIPO8 Class Constructor Function			
SIPO8	`uint8_t Max_SIPOs,` `uint8_t Max_timers`	n/a	The constructor function for the SIPO8 library class. Creates the internal structures required to map every output pin of the specified number of SIPO ICs. If memory cannot be mapped (insufficient free memory) the function will terminate after providing an error message to the serial monitor (set for 9600 baud).
SIPO Array Pool Functions (Absolute Addressing, *pin addresses - 0 to (number_active_pins_in_array -1)*)			
`void set_all_array_pins`	`bool pin_status`	n/a	Sets every active SIPO output pin to the given status, `LOW` or `HIGH`. It is equivalent to `set_banks(pin_status)`.
`void invert_all_array_pins`	none	n/a	Inverts the existing status of every active SIPO output pin. The function is equivalent to `invert_banks`.
`int set_array_pin`	`uint16_t pin,` `bool pin_status`	pin number, or `pin_set_failure`	Sets the specified SIPO output pin to the given status, `LOW` or `HIGH`. The pin must be an active pin. If successful returns the pin address, otherwise returns -1 (`pin_set_failure`).
`int invert_array_pin`	`uint16_t pin`	status of inverted pin, or `pin_invert_failure`	Inverts the existing status of the specified SIPO output pin. The pin must be an active pin. If successful returns the pin's new inverted status, otherwise returns -1 (`pin_invert_failure`).
`int read_array_pin`	`uint16_t pin`	status of pin, or `pin_read_failure`	Reads the status of the specified pin in the active array. If successful returns the pin's existing status, otherwise returns -1 (`pin_read_failure`).
`void xfer_array`	`bool msb_or_lsb`	n/a	Shifts out/xfers the entire array of active virtual SIPO output pins to the physically connected SIPO ICs. The function is equivalent to the function `xfer_banks(msb_or_lsb)`.
SIPO Bank Functions (Relative Addressing, *pin addresses - 0 to (number_active_pins_in_bank -1)*)			
`int create_bank`	`uint8_t data_pin,` `uint8_t clock_pin,` `uint8_t latch_pin,` `uint8_t num_SIPOs`	the bank's id, or `create_bank_failure`	Function creates a bank of SIPOs, assigning virtual output pins from the virtual pin array to the bank. Returns bank's id if successful, otherwise -1

Function/Method	Parameters	Returned Value(s)	Comments
			(`create_bank_failure`).
void set_bank	uint8_t bank, bool pin_status	n/a	Function sets every SIPO output pin defined by the bank to the given status, LOW or HIGH.
void set_banks	uint8_t from_bank, uint8_t to_bank, bool pin_status	n/a	Function sets every SIPO output pin in every bank in the given range of banks, to the given status, LOW or HIGH.
void set_banks	bool pin_status	n/a	Function set every SIPO output pin in every bank to the given status, LOW or HIGH. It is equivalent to set_all_array_pins(pin_status).
void invert_bank	uint8_t bank	n/a	Function will invert the existing pin status of every pin in the specified bank.
void invert_banks	uint8_t from_bank, uint8_t to_bank	n/a	Function will invert the existing pin status of every pin in each of the specified banks.
void invert_banks	none	n/a	Function will invert every SIPO output pin in every bank. Function is equivalent to invert_all_array_pins.
int set_bank_pin	uint8_t bank, uint8_t pin, bool pin_status	pin's absolute address, or pin_set_failure	Function sets the given pin (relative address) in the given bank to the given status, LOW or HIGH. Note that if successful, the return value is the pin's absolute address in the array, otherwise it is -1 (pin_set_failure).
int invert_bank_pin	uint8_t bank, uint8_t pin	pin's inverted status, or pin_invert_failure	Function inverts the given pin (relative address) in the given bank. Note that if successful, the return value is the pin's inverted status_, otherwise it is -1 (pin_invert_failure).
int read_bank_pin	uint8_t bank, uint8_t pin	status of pin, or pin_read_failure	Function reads the status of the specified pin in the specified bank. Note that if successful, the return value is the pin's status_, otherwise it is -1 (pin_read_failure).
int set_bank_SIPO	uint8_t bank, uint8_t SIPO_num, uint8_t SIPO_value	status_byte address, or SIPO_not_found, or bank_not_found	Function sets the specified SIPO (block of eight output pins) to the specified 8bit value. If successful the return value is the address of the status_byte holding the SIPO's 8bit status values, otherwise, if the specified SIPO does not exist in the bank -2 (SIPO_not_found), ortherwise -1 if the specified bank does not exist (bank_not_found).
int invert_bank_SIPO	uint8_t bank,	status_byte address,	Function inverts the specified SIPO (block of eight

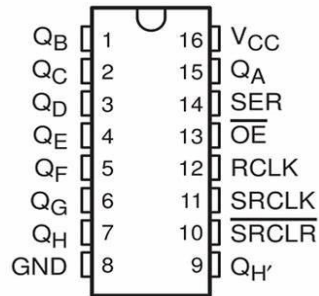| Function/Method	Parameters	Returned Value(s)	Comments

	`uint8_t SIPO_num`	`or`
		`SIPO_not_found`, `or`
		`bank_not_found`
			output pins). If successful the return value is the address of the status_byte holding the SIPO's 8bit status values, otherwise, if the specified SIPO does not exist in the bank -2 (`SIPO_not_found`), ortherwise -1 if the specified bank does not exist (`bank_not_found`).

`int  read_bank_SIPO`	`uint8_t bank,`
	`uint8_t SIPO_num`
		`status_byte value,`
		`or`
		`SIPO_not_found`, `or`
		`bank_not_found`
			Function reads the specified SIPO (block of eight output pins). If successful the return value is the value of the SIPO's status_byte (8bits), otherwise, if the specified SIPO does not exist in the bank -2 (`SIPO_not_found`), ortherwise -1 if the specified bank does not exist (`bank_not_found`).

Other SIPO Functions			
`int  get_bank_from_pin`	`uint16_t pin`	`bank_id`, `or` `bank_not_found`	Function returns the bank id the specified pin (absolute address) resides in. If the pin cannot be located in a bank the function returns -1 (`bank_not_found`).
`int  num_pins_in_bank`	`uint8_t bank`	`number of pins`, `or` `bank_not_found`	Function returns the number of SIPO pins the given bank maps/owns. If the bank id is not valid the function returns -1 (`bank_not_found`).
`void xfer_bank`	`uint8_t bank,` `bool msb_or_lsb`	n/a	The function shifts out/xfers all of the SIPO pins within the specified bank.

`void xfer_banks`	`uint8_t from_bank,`
	`uint8_t to_bank,`
	`bool msb_or_lsb`
		n/a	The function shifts out/xfers all of the SIPO [pins of each of the specified banks/

`void xfer_banks`	`bool msb_or_lsb`	n/a	The functions shifts out/xfers all of te SIPO pins for ALL banks. The function is equivalent to the `xfer_array`(msb_or_lsb) function.
`void print_pin_statuses`	none	n/a	A useful function for development/debugging – prints the pin status of every <u>active</u> SIPO pin in every bank.
`void print_SIPO_data`	none	n/a	A useful function for development/debugging – prints the SIPO8 library data set up by the class initiation process. The output confirms that the sketch's configuration is setup as intended.
`void SIPO8_start_timer`	`uint8_t timer`	n/a	The function marks the specified timer as active.
`void SIPO8_stop_timer`	`uint8_t timer`	n/a	The function marks the specified timer as inactive.
`bool`	`uint8_t timer,`	`elapsed` or	The function determines if the specified timer has

Ron Bentley, Stafford UK

Function/Method	Parameters	Returned Value(s)	Comments
SIPO8_timer_elapsed	uint32_t elapsed_time	!elapsed	reach/surpassed the given elapsed time parameter. The return value is true if timer has elapsed (elapsed), otherwise false (!elapsed). If the specified timer is not a valid timer the return value is also false (!elapsed).
Private Functions			
void shift_out_bank	uint8_t data_pin, uint8_t clock_pin, uint8_t status_bits, bool msb_or_lsb	n/a	A standard shift out function – transfers the given status_bits to the 3-wire digital interface defined for the associated bank. Function is called from the xfer functions.
void SIPO_lib_exit	uint8_t reason	n/a	The function is used exclusively by the class constructor if the required memory requirements cannot be met from the available free memory.

SIPO Mapping Table/Documentation

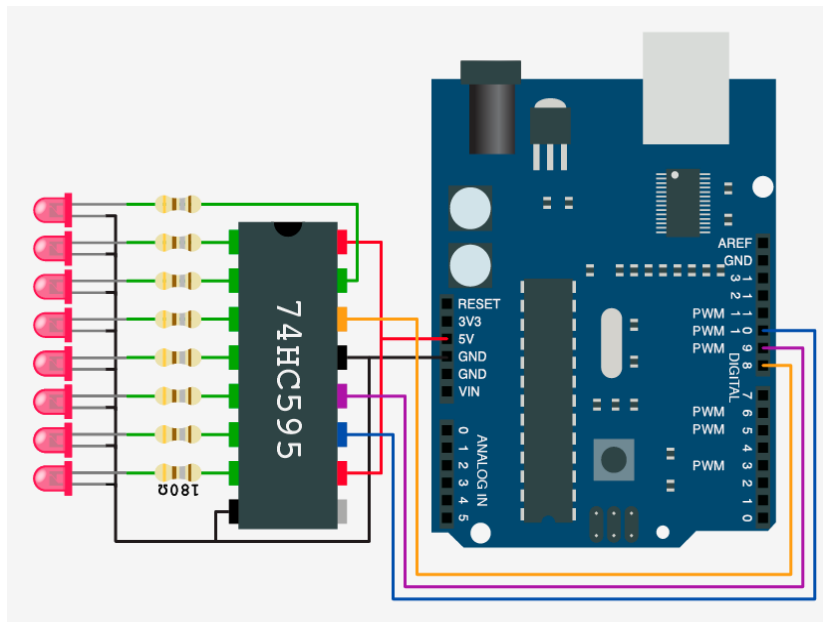| Project:							Date:	
User Defined Parameters				create_bank Results				
Control Pins			Num SIPO ICs in Bank	Bank Num	First Pin	Last Pin	Comments	
Data	Clock	Latch						

(add more rows as needed)

SIPO 8bit IC Pin Outs/Wiring Scheme

Pin Outs 1 - 74HC595

Example pin assignments	
74HC595 pins	Arduino pins
14, SER, DATA IN pin	8
12, RCLK, LATCH pin	9
11, SRCLK, CLOCK pin	10

Single SIPO IC
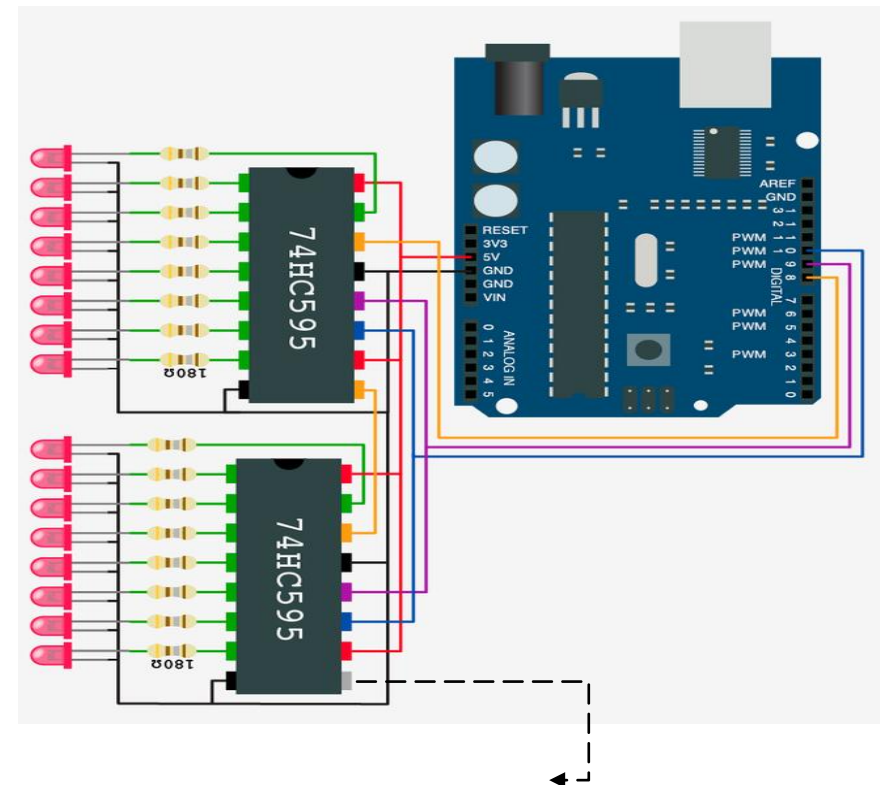
Dual SIPO ICs, cascaded,   next SIPO IC to pin 14

Ron Bentley, Stafford UK