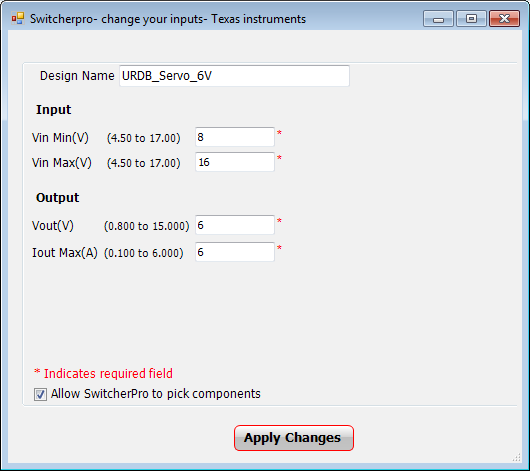
## Servo Power Block

Power is supplied to the servo connections by a TPS54620 switching regulator. By default, this regulator is configured to output 6V. RC servos use a variety of supply voltages, generally from 4.4V to 8V, with some larger servos operating off of as much as 12V. Changing the output voltage of the TPS54620 requires changing the value of many of the passive components in the regulator circuit. For ease of soldering, the relevant components are through-hole parts instead of the smaller 0603/0805 surface mount packages used elsewhere on the board. To calculate the correct component values for a desired output voltage, Texas Instruments provides a utility called SwitcherPro. Create a new design in SwitcherPro using the TPS54620 and set the inputs as shown below, changing the Vout value as desired.



Note: Do not use the exact parts recommended by SwitcherPro, as they are not the correct package sizes. The pads are sized for vertically-mounted 1/4W PTH resistors, ceramic capacitors with 5mm spacing, and vertically-mounted axial inductors. Values for common output voltages are shown below.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Table X: Servo power block component values** | | | | | |
| **Component (SwitcherPro)** | **Component (URDB label)** | **VO = 4.4V** | **VO = 6V** | **VO = 8V** | **VO = 12V** |
| R1 | R146 | 45.3K | 66.5K | 90.9K | 140K |
| R2 | R147 | 10K | 10K | 10K | 10K |
| R3 | R37 | 2.15K | 1.96K | 2.26K | 2.26K |
| R4 | R36 | 100K | 100K | 100K | 100K |
| R5 | R38 | 205K | 221K | 243K | 287K |
| R6 | R145 | 53.6K | 41.2K | 33.2K | 26.1K |
| C1 | Not Available | Open | Open | Open | Open |
| C2 | C52 | Open | Open | Open | Open |
| C3 | C51 | 100uF | 22uF | 22uF | 22uF |
| C4 | C50 | 0.018uF | 0.018uF | 0.018uF | 0.018uF |
| C5 | C16 | 4.7uF | 4.7uF | 4.7uF | 4.7uF |
| C6 | C34 | 0.022uF | 0.068uF | 0.047uF | 8200pF |
| C7 | C51 | 560pF | 68pF | 68pF | 390pF |
| C8 | C15 | 22uF | 22uF | 22uF | 22uF |
| L1 | L2 | 3.9uH | 4.7uH | 4.7uH | 3.3uH |

## XMOS-STM32 Communication Link

A device initiates communication by sending an opcode with the CMD/REPLY flag set to 0. The receiving device acknowledges the command by sending a reply with the same opcode. After the ACK byte is sent, both devices begin the transfer of data associated with the opcode. In the event of an opcode being added in the future calling for both devices to send data, these transfers can occur simultaneously. Neither device shall initiate a new command until the current exchange is completed.

### I/O Pin Message Data Format

Opcodes 35 and 36 pertain to the I/O pins connected to the Arduino headers of the STM32 coprocessor. All functionality of these STM32 I/O pins is exposed to the XMOS. When sending a GET\_IO\_PIN request, only the first 6 bits (the Arduino pin ID) need to be sent. When sending a GET\_IO\_PIN reply or a SET\_IO\_PIN request, the full 3-byte data packet must be sent.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 16 |
| Must be 0 | | | | | | | Value | AF[3] |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| AF [2:0] | | | PU/PD[1:0] | | Speed[1:0] | | Type | Mode[1:0] | | Pin ID[5:0] | | | | | |

|  |  |
| --- | --- |
| Bits 23:18: | Reserved, must be set to 0. |
| Bit 17: | **Pin value**  If Mode = 00: Pin’s IDR value.  If Mode = 01: Pin’s ODR value.  If Mode = 10 or 11: 0. |
| Bits 16:13 | **Pin alternate function**  Pin’s alternate function register (AFRL or AFRH) value. |
| Bits 12:11 | **Pin pull-up/pull-down configuration**  Pin’s PUPDR value. |
| Bits 10:9 | **Pin output speed**  Pin’s OSPEEDR value. |
| Bit 8: | **Pin type**  Pin’s OTYPER value. |
| Bits 7:6 | **Pin mode**  Pin’s MODER value. |
| Bits 5:0 | **Pin Arduino ID**  Numeric designation of the pin on the Arduino silkscreen. |

## XMOS Threads

The base URDB firmware executes six concurrent threads during device operation.. This leaves 24 free threads for application code.

|  |  |
| --- | --- |
| **Table X: XMOS threads** | |
| **Thread** | **Core** |
| IMU I2C Server | 3 |
| SPI Server | 0 |
| STM32 UART Server | 2 |
| PWM Module | 2 |
| Navigation Service | 3 |
| XMOS JTAG USB UART | 0 |

In addition to the base firmware threads, the following core assignments are required when using the Pmod and UEXT expansion headers. Note: Every I/O pin on these ports is an XMOS 1-bit port. While the hardware is configured with these expansion header formats in mind, they can be used as independent 1-bit ports by threads running on the same core if desired.

|  |  |
| --- | --- |
| **Table X: Expansion header threads** | |
| **Thread** | **Core** |
| Pmod A | 0 |
| Pmod B | 1 |
| Pmod C | 2 |
| UEXT I2C Server | 1 |
| UEXT UART Server | 1 |