

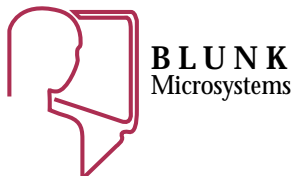
# TargetLAPB™

## LAPB PROTOCOL STACK

### HIGHLIGHTS

- Royalty Free
- ISO/IEC 7776 Compliant
- Supports Point-to-Point Networks
- Provides Flow Control and Reliability
- Configurable T1, T2, T4, N2, Window Size, Sequence Numbering, and Addressing
- Supports Multiple HDLC Channels
- Built-in LAPB Protocol Decoder
- Works with TargetOS
- Integrated with CodeWarrior

Blunk Microsystems provides system software, device drivers, and board support packages to the embedded systems market, both off-the-shelf products and custom work done under contract.



TargetLAPB is compliant with International Standard ISO/IEC 7776:1995(E) "Description of the X.25 LAPB-compatible DTE data link procedures" and has been extensively tested and demonstrated interoperable with other LAPB protocol stacks.

TargetLAPB supports exchanging data on point-to-point networks and provides automatic flow control and data reliability. Reliability is provided using CRC checks, frame sequence numbering, and positive acknowledgments with resends after timeouts. Flow control is provided using LAPB's sliding window protocol.

Configuration parameters T1, T2, T4, N2, window size, sequence number modulus (standard or extended), and address (DTE or DCE) are independently programmable for each channel.

LAPB events are reported to the application layer using an optional callback function. The following events are reported: N2 unacknowledged SABM/SABME commands, N2 unacknowledged DISC commands, N2 RNR responses, frame reject, physical disconnect, link reset, link reconnect, all I-frames acknowledged, and I-frame arrival.

Statistics are maintained on the number of connections/reconnections, number of frames sent and received, etc. The statistics may be read and/or written to `stdout` with the `lapbStatistics()` call.

Built-in LAPB protocol decoder. When enabled by a compile-time flag, a short summary of every packet transmitted or received is printed to `stdout` on an ongoing basis as the protocol stack runs.

TargetLAPB has a clearly documented HDLC driver interface and supports concurrent use of multiple channels and multiple serial controllers. HDLC drivers for Motorola MPC8XX, MC68360, and MC68302 SCC channels and other serial modules are available separately. Blunk Microsystems provides competitive bids on custom drivers.

To help catch errors during development, a compile-time option promotes TargetLAPB service call parameter errors to fatal errors that break into the command line monitor after printing an error message to `stderr`.

Shipped with a sample application designed to endlessly send I-frames to a LAPB test set operating in "echo" mode. All transmitted data is echoed back to the application, which tests the returning data for errors. The sample application is a good test of the TargetLAPB configuration, HDLC driver integration, and physical cabling.

Developed using TargetOS™, Blunk Microsystems' real time operating system, the source code is 100% ANSI C and is easily ported to other real-time kernels.

Integrated with CodeWarrior™, the development environment from Metrowerks with an integrated source level debugger, compiler, assembler, linker, editor, and GUI make tool.

Royalty free. Includes full source code, user's manual, sample application, and one year of technical support. ►

## CONTACT INFORMATION

- Visit our web site:  
[www.blunkmicro.com](http://www.blunkmicro.com)
- Customer Support:  
**(408) 323-9833**
- Technical Support:  
**(408) 323-1758**
- Fax:  
**(408) 323-1757**
- E-mail:  
[sales@blunkmicro.com](mailto:sales@blunkmicro.com)
- Address:  
**6576 Leyland Park Drive  
San Jose, CA 95120  
USA**

## AVAILABLE COMPONENTS

### TargetOS

Real-time, deterministic, multi-tasking, priority-based, preemptive kernel. Includes Standard C library.

### TargetTCP

RFC compliant TCP/IP protocol stack providing the standard Berkeley Sockets API.

### TargetFFS

POSIX compatible flash file system. Implements wear-leveling to prolong life of flash media. File system integrity is guaranteed across unexpected resets.

## The TargetLAPB Application Program Interface:

```
int lapbInit(void (*report)(int chan, ui32 event), void (*parse)(ui8 *data));
```

Initializes TargetLAPB and provides two optional call-back functions.

```
int lapbConfigure(int channel, LapbSpec *lapb_spec);
```

Configures a channel. LapbSpec defines the LAPB configuration.

```
typedef struct  
{  
    uint T1;           /* resend timeout (tenths of seconds) */  
    uint T2;           /* acknowledgment delay (tenths of seconds) */  
    uint T4;           /* idle channel timeout (tenths of seconds) */  
    uint N2;           /* max retry value */  
    uint window_size; /* max window size (less than 8/128) */  
    ui8 seq_modulus;  /* LAPB_SEQ_MOD_8 or LAPB_SEQ_MOD_128 */  
    ui8 addr_type;    /* LAPB_DTE_ADDR or LAPB_DCE_ADDR */  
} LapbSpec;
```

```
int lapbConnect(int channel, int send_sabm);
```

Establishes a channel's LAPB connection.

```
int lapbDisconnect(int channel, uint flags);
```

Terminates a channel's LAPB and/or physical connection.

```
int lapbSend(int channel, void *buffer, int len);
```

Transmits an application buffer as the data field in a LAPB I-frame.

```
int lapbRecv(int channel, void *buffer);
```

Receives data from a LAPB channel.

```
int lapbStatistics(int channel, int print, LapbStats *lp, PhysStats *pp);
```

Accesses a channel's statistics. If print is TRUE, a report is written to stdout. ■

## Licensing Terms

TargetLAPB™ is royalty free. Purchasers are granted a non-exclusive license to use the provided source code at a single site. Licensees have the right to disseminate or resell the software in executable format only. The source code and derived object code may not be redistributed or resold.