Enhanced WWVB Broadcast Format

John Lowe (john.lowe@nist.gov) Time and Frequency Services National Institute of Standards and Technology

December 7, 2012

1. Introduction

The WWVB broadcast of the official time of the US government has existed since 1965. Since then, NIST has upgraded the broadcast system and modified the signal several times making the service more accessible to the public, and resulting in large numbers of radio-controlled clocks (RCCs). The most recent improvements included a significant increase in broadcast power and an increase in the modulation factor used for the amplitude-modulation, both of which served to improve reception coverage for existing RCCs and increase their reliability. Nevertheless, it has been realized that these popular devices still often encounter difficulties in reception, which depend on the geographical location, time of day, type of structure, and interference sources that may be present in a given environment. All of these factors determine what is called the signal-to-noise-and-interference-ratio (SNIR) that a receiver experiences at a given instance.

In order to address the reception challenges and provide the public with a much improved system, NIST is now introducing an enhanced communication protocol, to which phase modulation was added, offering significantly improved performance in new products that are designed according to the new protocol. The new protocol maintains the amplitude-modulation (AM) and pulse-width modulation (PWM) of the legacy protocol, the details of which have been made available in NIST Special Publication 250-67 (<u>http://tf.nist.gov/general/pdf/1969.pdf</u>) from 2005, where many additional information about the WWVB station may be found. This backward compatibility ensures that typical existing consumer-market products, based on conventional envelope detection, are not impacted. This means that although legacy receivers cannot benefit from these improvements, their performance will not be degraded, either.

However, receivers designed to lock to the carrier's phase and perform coherent detection, which are typically common only in more-professional equipment, are impacted by the introduction of the phase-modulation defined in the protocol. Based on the early notification provided by NIST and trials that were performed throughout 2012, it is expected that these receivers will either be modified or replaced. In the transitional period, to extend into 2013, the phase modulation will be disabled for 30 minutes twice a day, at noon and at midnight Mountain Standard Time (MST), allowing carrier-locked based time-keeping devices to resynchronize to the broadcast in its legacy form (i.e., having only amplitude and pulse-width modulations).

This document specifies the data content, physical properties and scheduling features of the phasemodulating (PM) time code that has been added to the WWVB broadcast. It is intended to allow users to correctly interpret the various components in the PM code. It should be noted that there are differences between the information made available through the PM code and what has been available through the legacy AM/PWM protocol. For example, while the time and date may be extracted from both, the leap year indication is not duplicated in the PM code, whereas the PM code contains a new field that provides advance notification for daylight-saving time (DST) transitions.

Additional features, offering further enhancements to the user experience in various SNIR conditions, will be described in a future public release in 2013.

2. General Properties of the Phase Modulation (PM) Protocol

The signal properties of the new broadcast are designed to maintain backwards compatibility with the common envelope detector-based receivers that were designed to operate with the legacy AM/PWM WWVB protocol. These receivers, found in many low-cost consumer market products, are typically based on a crystal filter centered at 60 kHz and having a bandwidth narrower than 10 Hz, which is followed by a non-linear envelope detection operation (rather than a coherent detector, which is based on multiplication with a locally generated 60 kHz signal that is phase-locked to the modulated carrier).

The PM protocol was designed to allow for flexibility/scalability (i.e., optimized operation at a very wide range of SNR values), while also making provisions for additional features and extensions. It is anticipated that details for additional features will be published before the end of the year. These features will allow faster and more accurate synchronization, as well as address the problems of receivers with particularly low SNR.

2.1. Definition of the Phase Modulation

The PM format is based on antipodal binary phase shift keying (BPSK), i.e., the two symbols are 180° apart. A "0" is represented by the carrier's non-modulated phase, as with the phase modulation turned off, whereas a "1" is represented by an inverted carrier. The hourly 45° phase shift that had existed in the legacy broadcast for station identification is eliminated, as station identification becomes possible based on the many unique signatures in its new PM code, detailed in the following sections, which distinguish it from other broadcasts.

2.2. Physical Properties of the Modulating Baseband Signal

As can be seen in Figure 1, the baseband signal, which combines the two-level legacy AM/PWM signal and the phase (sign) inversions, may experience at least four different levels in a phase-modulated frame. These correspond to the legacy AM levels V_H and V_L , having the ratio $V_H/V_L \cong 7$, each of which may be multiplied by either a +1, representing a "0" in the BPSK modulation, or -1 for phase reversal, representing a "1" in the BPSK modulation.

The phase transition between each bit and the next one in the 1 bps (bit per second) PM frame occurs 100 ms after the AM amplitude drop that indicates the end of that second, as shown in Figure 1, illustrating an example baseband version of a transmitted symbol, where the information in the PM is shown to transition from a "0" to "1", while the transmitted AM bit is a "1". The baseband signal shown in this figure is multiplied by the 60kHz carrier in the transmitter, thereby resulting both in variations in the carrier's amplitude and in sign reversals in it whenever the baseband signal assumes negative values.

Figures 2-5 illustrate the modulated carrier for all of 4 combinations of 0/1 bit values for the legacy and the PM frames (markers not shown).







Figure 2 – The modulated carrier for a "0" both in the legacy protocol and in the PM



Figure 3 – The modulated carrier for a "0" in the legacy protocol and a "1" in the PM



Figure 4 – The modulated carrier for a "1" in the legacy protocol and a "0" in the PM



Figure 5 – The modulated carrier for a "1" both in the legacy protocol and in the PM

Since the lower-power portion of each bit, transmitted with an amplitude of V_L , is about 50 times weaker in power than the high-power portion, it is recommended that the phase of the carrier for each data bit be determined in the receiver based only on its high-power interval.

3. Scheduling of Phase-Modulated Frames

The 60-second phase modulated time information frames coincide with the legacy AM/PWM 60-second frames. However, while the legacy AM/PWM modulation is present 100 % of the time, the normal phase modulation on the carrier may be briefly interrupted for various purposes.

During the transitional period, to extend at least until the end of 2012, PM interruptions for durations of 30 minutes will be scheduled to occur daily at noon and at midnight MST, to allow for existing carrier-locking based equipment to resynchronize or relock to the legacy AM/PWM signal.

Messages for various purposes, such as emergency notices, may be incorporated into the broadcast and scheduled to specific instances. A message frame, transmitted at the rate of 1 bps, replaces a normal time frame. Therefore, it may not be used for time acquisition in a phase-modulation based RCC. However, it may still be used for timing correction (tracking), because it starts with a known synchronization word, as detailed in the next section. A message may also exceed a single frame and span over multiple frames. The broadcasting of messages will be limited to a low duty cycle (e.g., below 10 % of the time), particularly during times assumed to be most critical for reception in RCC devices.

4. Bit Allocation and Notation in 1 bit/s Time Frame

The PM bit allocation for a 60-bit 1 bps frame dedicated to time information is described in Table 1 against the bit allocation of the legacy AM/PWM protocol. This section describes the contents and significance of the various fields in the one-minute time frame, while Section 5 and Table 9 describe the message frames, which may infrequently replace time frames in the PM signal.

Note that bits in N-bit words are numbered from 0 for the least-significant bit (LSB) to N-1 for the most significant bit (MSB), with the MSB being transmitted first. For example, the 26-bit time word (minute counter) is numbered from 0 (LSB) to 25 (MSB), with the 25th bit (MSB) being transmitted first.

Second	0	1	2	3	4	5	6	7	8	9
Legacy AM/PWM	Marker	min_40	min_20	min_10	0	min_8	min_4	min_2	min_1	Marker
Phase	sync_T[12]	sync_T[11]	sync_T[10]	sync_T[9]	sync_T[8]	sync_T[7]	sync_T[6]	sync_T[5]	sync_T[4]	sync_T[3]
Second	10	11	12	13	14	15	16	17	18	19
Legacy AM/PWM	0	0	hour_20	hour_10	0	hour_8	hour_4	hour_2	hour_1	Marker
Phase	sync_T[2]	sync_T[1]	sync_T[0]	time_par[4]	time_par[3]	time_par[2]	time_par[1]	time_par[0]	time[25]	time[0]
Second	20	21	22	23	24	25	26	27	28	29
Legacy AM/PWM	0	0	day_200	day_100	0	day_80	day_40	day_20	day_10	Marker
Phase	time[24]	time[23]	time[22]	time[21]	time[20]	time[19]	time[18]	time[17]	time[16]	R
Second	30	31	32	33	34	35	36	37	38	39
Legacy AM/PWM	day_8	day_4	day_2	day_1	0	0	UT1_S[2]	UT1_S[1]	UT1_S[0]	Marker
Phase	time[15]	time[14]	time[13]	time[12]	time[11]	time[10]	time[9]	time[8]	time[7]	R
Second	40	41	42	43	44	45	46	47	48	49
Legacy AM/PWM	UT1_C_0.8	UT1_C_0.4	UT1_C_0.2	UT1_C_0.1	0	year_80	year_40	year_20	year_10	Marker
Phase	time[6]	time[5]	time[4]	time[3]	time[2]	time[1]	time[0]	dst_ls[4]	dst_ls[3]	notice
Second	50	51	52	52	54	55	56	57	59	50
	JU Voar 8	JI Voar 4	JZ Voar 2	voar 1	0			DST[1]		Markor
Legacy AIVI/F WIVI	yeal_o	yeal_4	yeal_2	yeal_1	0		1.500			IVIAI KEI
Phase	ast_is[2]	ast_is[1]	ast_is[0]	dst_next[5]	ast_next[4]	ast_next[3]	ast_next[2]	ast_next[1]	dst_next[0]	0

Table 1 - Allocation of bits in one-minute time frame (legacy protocol in black, PM in red)

- The significance of each of the fields in the legacy AM protocol is given in previous NIST publications (referenced in the Introduction section) and is not repeated here.

- Bits 29 and 39, designated "R", are reserved for future use and have not been assigned at this time.

- Bits 4, 14, 24, 34, 44 and 54 in the legacy AM system were historically reserved for future use, but are now set permanently to 0.

4.1. Listing of the Fields in a Time Frame

Table 2 lists the six different fields in a time frame, which add up to 60 s in duration.

Since the duration of the high power level in markers, as defined by the legacy protocol, is only 200 ms (compared to 500 ms and 800 ms for the "1" and "0" symbols respectively), none of the time information bits rely on a marker. The purpose of each field is described in the subsequent subsections.

	purpose of field	bits allocated	total number of bits (and seconds in duration)
1	synchronization word (may include last bit of previous frame)	0-12, 59	14
2	time word (includes 5 parity bits and repeated LSB)	13-28, 30-38, 40-46	32
3	daylight saving time (DST) state and leap second notification	47-48, 50-52	5
4	advance notice for next DST transition (or message word)	53-58	6
5	NIST notice indication	49	1
6	reserved bits (coincide with markers in AM/PWM)	29, 39	2
		total:	60

Table 2 - List of fields in a one-minute time frame

4.2. Synchronization Word

The known 13-bit synchronization word, {**sync[12], sync[11],... sync[0]**}, populating bits 0-12, is used for the purpose of timing (marks the beginning of a minute) and conveys no information. The last bit in every frame (bit 59, which is a marker in the AM frame), is always 0, and may be considered an extension to the sync word, extending it to a total of 14 seconds (i.e., the last bit from a previous frame may be appended to the first 13 bits in the next frame). One of two synchronization words may be used, denoted **sync_T** and **sync_M**, for time frames and message frames respectively, as specified in Table 3.

bit #	12	11	10	9	8	7	6	5	4	3	2	1	0
sync_T	0	0	1	1	1	0	1	1	0	1	0	0	0
sync_M	1	1	0	1	0	0	0	1	1	1	0	1	0

Table 3 - 13-bit synchronization words in time and message frames

4.3. Time Word

The time word, from which the year (excluding century), the date, and the hour and minutes may be extracted, is represented by a 26-bit minute counter that is reset at the turn of the century, i.e., the current value in it should be considered to represent the number of minutes that have elapsed since this century began at 00:00 UTC on January 1st 2000.

For example, as the time turns from 21:29:59 UTC to 21:30:00 UTC on July 28th, 2016, the minute counter will be incremented from 8717609 to 8717610 (decimal). The binary representation of 8717610 will appear within the 60 s frame that starts at that instance (i.e., the 21:30:00 UTC instance is announced after it has already occurred). This timing of the frame contents with respect to the beginning of the minute frame (the "on time" mark at the beginning of the first second of the minute) is identical to the convention defined in the legacy protocol.

The 26-bit time word is encoded into a 31-bit code-word, by calculating five additional parity bits that are appended to it because of the error-correcting linear block code Hamming(31, 26), based on the equations provided below. This code provides the receiver with the capability to correct one error and to detect up to two errors in the 31-bit word.

- time_par[0] = sum(modulo 2){time[23, 21, 20, 17, 16, 15, 14, 13, 9, 8, 6, 5, 4, 2, 0]}
- time_par[1] = sum(modulo 2){time[24, 22, 21, 18, 17, 16, 15, 14, 10, 9, 7, 6, 5, 3, 1]}
- time_par[2] = sum(modulo 2){time[25, 23, 22, 19, 18, 17, 16, 15, 11, 10, 8, 7, 6, 4, 2]}
- time_par[3] = sum(modulo 2){time[24, 21, 19, 18, 15, 14, 13, 12, 11, 7, 6, 4, 3, 2, 0]}
- time_par[4] = sum(modulo 2){time[25, 22, 20, 19, 16, 15, 14, 13, 12, 8, 7, 5, 4, 3, 1]}

The sum(modulo 2) operation is equivalent to an exclusive OR logic operation.

The 31-bit word is placed in the time frame in locations {time_par[4], time_par[3],...time_par[0], time[25], time[24],...time[0]}, corresponding to bits 13-28, 30-38, and 40-46, as shown in Table 1. The LSB, bit time[0], is repeated on bit 19, which is a marker in the legacy protocol (i.e., the duration of its high power portion is only 200 ms).

4.4. Indications for Daylight Saving Time (DST) and Leap Second

The 5-bit word {dst_ls[4], dst_ls[3]... dst_ls[0]} in locations 47-48 and 50-52 is used to indicate whether DST is in effect or not or whether it is starting or ending today, and is also used to indicate whether a positive or negative leap-second is scheduled at the end of this month (at minute 23:59 UTC). These four different possible indications for the DST state and the three for the leap second are all merged into a single 5-code word that allows for the detection of errors, particularly for the two most common combinations to be found in this 5-bit field (highlighted in Table 4). Table 4 maps each of the 12 legitimate values for the 5-bit dst_ls word into the corresponding values of the 2-bit indication dst_on and the 2-bit indication leap_sec. Table 5 and Table 6 list the significances for the various values of these two fields respectively.

	0	OST and lea	ap second	code wor	d	significance (DST and leap second bits)					
	dst_ls[4]	dst_ls[3]	dst_ls[2]	dst_ls[1]	dst_ls[0]	dst_on[1]	dst_on[0]	leap_sec[1]	leap_sec[0]		
1	0	1	0	0	0	0	0	0	х		
2	1	0	1	1	0	1	0	0	х		
3	0	0	0	1	1	1	1	0	x		
4	1	0	1	0	1	0	1	0	х		
5	0	0	1	0	0	0	0	1	0		
6	1	0	0	0	0	1	0	1	0		
7	0	1	1	0	1	1	1	1	0		
8	0	1	1	1	0	0	1	1	0		
9	1	1	0	0	1	0	0	1	1		
10	1	1	0	1	0	1	0	1	1		
11	1	1	1	1	1	1	1	1	1		
12	1	1	1	0	0	0	1	1	1		

Table 4 - Decoding table for DST and leap second indication word {dst_ls[4], dst_ls[3]... dst_ls[0]}

Bits {dst_on[1], dst_on[0]}, which are to be extracted from the dst_ls word according to the decoding scheme provided by Table 4, indicate the DST state, as shown in Table 5 and explained as follows: Bit dst_on[1] is set to 1 at 00:00 UTC on the first Sunday of the DST period (in the spring) and is reset at 00:00 UTC on the last Sunday ending the DST period (in the fall). It is to be noted that since 00:00 UTC occurs a few hours before midnight in all time zones in the United States, and the DST transitions are to be implemented after midnight, the transitions in the dst_on[1] bit occur a number of hours before a receiving device is required to make the appropriate 1-hour correction, depending on which time zone the device is in. For this reason, the second bit, dst_on[0], serves to identify the period of time in which dst_on[1] has indicated that the DST period has started (or ended), but this change is not to take effect yet, since the time for that (currently set at 2 AM on a specific Sunday) has not yet been reached.

In the absence of this second bit, a device that first receives the time in the afternoon/evening on the last Saturday of the DST period (or the last one before a DST period starts), after 00:00 UTC, which is still before 6PM PST, for example, might incorrectly apply the DST time change a few hours prematurely.

Bit dst_on[0] follows the transitions on bit dst_on[1] with a delay of 24 hours (i.e., at midnight UTC between Sunday and Monday), in alignment with the corresponding 2-bit indication of the legacy AM/PWM protocol. The state of dst_on[0] allows the receiver to determine whether the DST state indicated by dst_on[1] has been valid for over a day, in which case the appropriate time correction is to be implemented immediately. If dst_on[0] indicates that the first day since the last DST transition is not over yet, then the RCC should not apply the new DST state in its time calculation until the appropriate time is reached (currently set at 2 AM in the local time zone).

dst_on[1]	dst_on[0]	significance
0	0	DST has not been in effect for over a day \rightarrow Apply standard time correction with respect to UTC. Next transition is into DST
1	0	DST starts today \rightarrow Do not apply the 1 hour DST correction until the appropriate time has been reached. *
1	1	DST has been in effect for more than a day \rightarrow Apply one hour less of time correction with respect to UTC. Next transition is out of DST.
0	1	DST ends today \rightarrow Continue to apply the DST correction and return to standard time only once the appropriate instance has been reached. *

Table 5 - DST state/transition word {**dst_on[1]**, **dst_on[0**]}

* The combinations 01 and 10 can only be present for 24 hours (during the entire Sunday of the transition).

Bit **leap_sec[1]**, when set to 1, indicates that a leap second is scheduled at the end of the current month, as indicated in Table 6. At 00:00 UTC at the beginning of each month its value is reset to zero if no leap second is scheduled for that month (i.e. all minutes will be of 60 seconds), or is set to 1 at that instance if the last minute of that month is to be either extended to 61 seconds or shortened to 59 seconds. Historically, only the months of June and December have been occasionally extended by one second, but the broadcast format can accommodate a leap second in any month, which may be positive or negative.

Table 6 - Leap second advance notification bit leap_sec

leap_sec[1]	leap_sec[0]	significance
0	х	no leap second at the end of this month
1	0	negative leap second scheduled for the last minute of this month
1	1	positive leap second scheduled for the last minute of this month

For the case of a positive leap second, the time frame representing the extended 61-second minute, starting at 23:59:00 UTC, will have bit 59 repeated (a marker in the legacy broadcast and a "0" in PM), after which bit **leap_sec[1]** will be reset to 0. For the case of a negative leap second the time frame representing the shortened minute will have bit 59 removed, such that bit 0 of the next minute (i.e. minute 00:00 UTC of the first day of next month) will follow immediately after bit 58.

The timing of minute-frame 00:00 UTC on the first day of the month following the implementation of the leap second, as well as the timing of the subsequent frames, will reflect the corrected time, and no further indication will be found in them for the leap second that has been implemented (the UT1 time correction information that is available in the legacy frame in the AM/PWM modulation is not provided in the PM signal).

4.5. Notice Bit

Bit 49, which is a marker in the legacy AM/PWM signal, indicates when a notice from NIST is posted at the website, as specified in Table 7. When set to 1, it may indicate that a change is imminent, a temporary shutdown is planned, or whatever other message, the details of which may be found at <u>http://www.nist.gov/pml/div688/grp40/wwvb.cfm</u>. It is expected that most consumer market products will ignore this bit.

Notice	significance
0	nothing to report
1	notice regarding WWVB broadcast may be found at NIST website

Table 7 - Notice Bit (bit 49)

4.6. Advance Notification for Next DST Transition

Bits 53-58, {dst_next[5], dst_next[4]...dst_next[0]}, usually represent one of eight possible schedules for the upcoming DST transition (i.e., either when the DST period is to start or end), but may also convey one of 8 other possible messages, as specified in Table 8. When DST is in effect, as is the case during the summer, the DST schedule word provides advance notification for the end of the DST period in the fall, whereas when DST is not in effect, as is the case in the winter, the DST schedule word provides advance notification for the beginning of the next DST period in the upcoming spring.

The start/end times for the DST period are always during the night between a Saturday and a Sunday, but the specific Sunday has changed over the years and may change again in the future.

As can be seen in Table 8, 24 combinations of day and time are supported for possible start/end times for the DST period, resulting in a total of 48 different 6-bit combinations. Historically, the DST transition in the US is to be implemented at 2 AM (local time), but combinations have been reserved for the possibility of this time being set at 1 AM or 3 AM, as is done in other countries. A total of eight Sundays are covered by the 24 combinations, both for the beginning as well as for the end of the DST period.

For combinations 49-56 the **dst_on** state has no relevance, as these do not represent specific start/end times. Message 49 is reserved for the case of a new DST schedule being defined that is not among the 24 predefined ones, message 50 is reserved for the possibility of DST being cancelled, while maintaining standard time, and message 51 has been reserved for the case of DST being permanently on (i.e., ahead of standard time by one hour). Additionally, five different words (messages 52-56) have been reserved, for which specific messages may be defined in the future, such as emergency messages.

(1) (or reserved message) of next DST transition 1 0 1 1 0 0 1 <th></th> <th>dst on</th> <th>DST</th> <th>۲ sche</th> <th>dule</th> <th>word</th> <th>dst_n</th> <th>ext</th> <th colspan="6">time and action for implementation</th>		dst on	DST	۲ sche	dule	word	dst_n	ext	time and action for implementation					
I - 1 [1] [2] </td <td></td> <td>[1]</td> <td></td> <td>or res</td> <td>serve</td> <td>d mes</td> <td>sage</td> <td></td> <td>of next DST trar</td> <td>nsition</td>		[1]		or res	serve	d mes	sage		of next DST trar	nsition				
1 0 1		[-]	[5]	[4]	[3]	[2]	[1]	[0]	day	time (local)				
2 0 1 0 0 1 1 0 2.1 dS unday of March after 0.59AM, 4 0 0 1 0 1 4.4 bunday of March skip from 5 0 1 1 1 1 0 4.4 bunday since "M" 3.00AM to 5 0 1 1 1 0 4.4 bunday since "M" 2:00AM 7 0 1 1 1 0 1.5 bunday since "M" 3:00AM 10 0 1 1 1 0 1.5 bunday since "M" 3:00AM 11 0 1 1 1 0 3:00AM 3:00AM 12 0 0 0 1 1 1 1 3:00AM 13 0 0 0 1 <td>1</td> <td>0</td> <td>1</td> <td>1</td> <td>0</td> <td>0</td> <td>0</td> <td>1</td> <td>1st Sunday of March</td> <td></td>	1	0	1	1	0	0	0	1	1st Sunday of March					
3 0 1 0 1 3/d Sunday of March 5 0 1 1 1 1 1 100 1 1000 1000 100 100	2	0	1	0	0	1	1	0	2nd Sunday of March	after 0.59AM				
4 0 0 1 0 1 4th Sunday of March 1:00 MM tot 5 0 1 1 1 1 4th Sunday since "M" 3:00 MM tot 6 0 0 1 1 1 1 6th Sunday since "M" 3:00 AM tot 7 0 1 1 0 1 1 0 1:00 Ath Sunday since "M" 3:00 AM tot 9 0 1 0 1 1 0 1:10 Ath Sunday since "M" 3:00 AM tot 11 0 0 0 1 1 0 1:10 Ath Sunday since "M" 3:00 AM tot 12 0 0 0 1 1 0 1:10 Ath Sunday since "M" 3:00 Ath tot 13 0 0 0 1 1 0 1:10 Ath Sunday since "M" 3:00 Ath tot 14 0 0 1 1 0 1:10 Ath Sunday since "M" 3:00 Ath tot 15 0 1 1 0 1:11 Ath Sunday of March 3:00 Ath tot 18 0	3	0	1	0	0	1	0	1	3rd Sunday of March	ckin from				
3 0 1 1 1 0 4 us sunday since "M" 1:00AW to 7 0 1 1 0 1 1 0 1:00AW to 8 0 1 1 0 1 1 0 1:00AW to 9 0 1 0 1 1 0 1:10AW to 2:00AM 10 0 0 1 1 0 1:10AW to 2:00AW 11 0 0 0 1 1 0:13 ds unday of March skip from 12 0 0 0 1 0 0 1:50AW to 3:00AW 14 0 0 0 1 0 0 1:55AW to 3:00AW 15 0 0 0 1:1 0 0 3:00AW to 3:00AW 16 1 0 1 0 0 1:1 Stawady of March 3:00AW skip from 22 0 1 1 0 0 1:1 Stawady of March 3:00AW 4fer 2:	4	0	1	1	1	1	1	1	4th Sunday of March					
0 0 1	5	0	1	1	1	1	1	0	5th Sunday since "M"	1:00AM to				
1 0 1	7	0	1	1	0	1	1	1	6th Sunday since "M"	2:00AM				
9 0 1 0 1 0 1st Sunday of March 10 0 0 1 1 0 1 1 2nd Sunday of March 11 0 0 0 1 1 0 3rd Sunday of March after 1:59AM, 12 0 0 0 0 1 1 0 3rd Sunday of March 3:00AM 13 0 0 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 1 0 1 1 0 1 1 0 1 1 0 1 1 0 1 <th1< th=""> <th1< th=""> <th1< th=""></th1<></th1<></th1<>	8	0	1	1	1	1	0	1	7th Sunday since "M"					
10 0 1 1 2 11 00 11 11 10 3rd Sunday of March after 1:59AM, 11 0 0 0 1 1 1 0 3rd Sunday of March skip from 13 0 0 0 0 1 0 4th Sunday of March skip from 13 0 0 0 1 0 4th Sunday of March 3:00AM 15 0 0 0 1 0 1 6th Sunday of March 3:00AM 16 0 1 0 1 0 0 3rd Sunday of March 3:00AM 17 0 0 1 1 0 0 3tfter 2:59AM, skip from 20 0 1 1 0 0 3tfter 2:59AM, skip from 21 0 1 1 0 0 3tfter 2:59AM, skip from 22 0 1 1 0 0 3tfter 1:59AM, skip from 22 1 1	9	0	1	0	1	0	1	0	1st Sunday of March					
11 0 0 0 1 1 1 0 3rd Sunday of March skip from 12 0 0 0 0 1 4th Sunday of March skip from 13 0 0 0 1 0 4th Sunday since "M" 2:00AM to 14 0 0 0 1 0 1 6th Sunday since "M" 3:00AM 15 0 0 1 0 0 1 6th Sunday of March 3:00AM 18 0 1 0 0 0 2rd Sunday of March 3:00AM to 20 0 1 0 0 0 3:00AM to 4:00AM 21 0 1 1 0 0 1 1 4:00AM 23 0 1 1 1 4th Sunday of Norember 3:00AM to 23 1 1 1 1 4th Sunday of Norember 3:00AM 24 0 1 1 1 1 1 1 1 1 1	10	0	0	1	1	0	1	1	2nd Sunday of March					
12 0 0 0 0 1 4th Sunday of March 2:00AM to 13 0 0 0 1 0 4th Sunday since "M" 2:00AM to 14 0 0 0 1 0 0 5th Sunday since "M" 2:00AM to 15 0 0 0 1 0 0 5th Sunday since "M" 3:00AM 16 0 1 0 0 1 th Sunday since "M" 3:00AM 17 0 0 0 1 0 1 st Sunday of March 3:00AM to 18 0 1 0 0 3:0'St Sunday of March 3:0OAM to 20 0 1 1 0 0'St Sunday of March 3:0OAM to 21 0 1 1 0'St Sunday for March 3:0OAM to 22 0 1 1 0'St Sunday for March 3:0OAM to 23 1 1 1 1'St Sunday for Movember 3:0OAM to 23 1 1 0'St Sunday for Movember 3:0OAM 0:0OA	11	0	0	0	1	1	1	0	3rd Sunday of March	after 1:59AM,				
13 0 0 0 1 0 4th Sunday since "M" 2:00AM to 14 0 0 0 1 0 0 5th Sunday since "M" 3:00AM 15 0 0 1 1 0 1 6th Sunday since "M" 3:00AM 16 0 1 0 0 1 6th Sunday since "M" 3:00AM 17 0 0 0 0 1 0 1 6th Sunday since "M" after 2:59AM, 18 0 1 1 0 0 3:00AM to 4:00AM 4:00AM 21 0 1 1 0 0 3:dt Sunday since "M" 3:00AM 5:0AM 5:0AM 3:0AM 4:00AM 4:00AM 4:00AM 4:00AM 4:00AM 4:00AM 6:0AM 4:0AM 6:0AM 6:0AM 6:0AM 6:0AM 6:0AM 6:0AM 6:0AM <td< td=""><td>12</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>4th Sunday of March</td><td>skip from</td></td<>	12	0	0	0	0	0	0	1	4th Sunday of March	skip from				
14 0 0 0 1 0 0 0 3:00AM 15 0 0 1 0 1 0 16t Sunday since "M" 3:00AM 16 0 1 0 1 0 15t Sunday of March 3:00AM 17 0 0 0 1 0 0 14t Sunday since "M" 4fter 2:59AM, skip from 19 0 1 1 0 0 4th Sunday since "M" 3:00AM to 20 0 1 1 0 0 4th Sunday since "M" 3:00AM to 21 0 1 1 0 0 4th Sunday since "M" 4:00AM 23 0 1 1 0 0 1.0 6th Sunday since "M" 3:00AM to 24 0 1 1 0 1.1	13	0	0	0	0	0	1	0	4th Sunday since "M"	2:00AM to				
15 0 0 0 1 1 0 1 1 0 1 1 0 1 1 0 1 1 0 1 1 0 1 1 0 1 1 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 1 0 1 1 1 0 1 1 1 0 0 1 <th1< th=""> <th1< th=""> <th1< th=""></th1<></th1<></th1<>	14	0	0	0	1	0	0	0	5th Sunday since "M"	3:00AM				
10 0 1 0 0 1 10 11 10 10 11 10 10 11 11 10 10 11 11 10 10 11 11 10 11 </td <td>15</td> <td>0</td> <td>1</td> <td>0</td> <td>1</td> <td>1</td> <td>0</td> <td>1</td> <td>7th Sunday since IVI</td> <td></td>	15	0	1	0	1	1	0	1	7th Sunday since IVI					
11 0 0 0 1 10 131 131 after 2:59AM, 19 0 1 1 0 0 20 0 1 10 11 0 0 3rd Sunday of March skip from 20 0 1 0 1 1 0 0 4th Sunday of March skip from 21 0 1 1 0 0 0 4th Sunday since "M" 3:00AM to 22 0 0 1 1 0 0 4th Sunday since "M" 3:00AM to 23 0 1 1 0 0 7th Sunday since "M" after 0:59AM, 24 0 1 1 0 1 </td <td>10</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>1</td> <td>0</td> <td>1</td> <td>1st Sunday of March</td> <td></td>	10	0	0	0	0	1	0	1	1st Sunday of March					
19 0 1 0 0 3rd Sunday of March after 2:59AM, 20 0 1 0 1 0 0 3rd Sunday of March skip from 21 0 1 1 1 0 0 4th Sunday since "M" 3:00AM to 22 0 0 1 1 0 0 5th Sunday since "M" 3:00AM to 23 0 1 1 0 0 7th Sunday since "M" 3:00AM to 24 0 0 1 1 0 7th Sunday since "M" after 0:59AM, 25 1 1 1 0 1 1 4th Sunday of November after 0:59AM, 26 1 0 1 0 1 1 0 1.100 <td>18</td> <td>0</td> <td>1</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>2nd Sunday of March</td> <td></td>	18	0	1	0	0	0	0	0	2nd Sunday of March					
20 0 1 1 0 0 4th Sunday of March 21 0 1 1 0 0 4th Sunday of March 22 0 0 1 0 0 4th Sunday since "M" 3:00AM to 23 0 1 1 0 0 5th Sunday since "M" 4:00AM 24 0 0 1 1 0 0 7th Sunday since "M" 4:00AM 25 1 1 1 0 0 1 3th Sunday before "N" after 0:59AM, instead of 26 1 0 1 0 1 1 1 1 1 26 1 0 1 0 1 3th Sunday of November after 0:59AM, instead of 30 1 1 1 1 1 1 1 0 1 3th Sunday of November 31 1 0 0 1 1 1 1 1 0 1 1 1 0 1 1:00AM 0:00AM 0:00AM<	19	0	1	1	0	1	0	0	3rd Sunday of March	after 2:59AM,				
21 0 1 1 1 0 0 4th Sunday since "M" 3:00AM to 22 0 0 1 0 0 0 5th Sunday since "M" 4:00AM 23 0 1 1 0 0 1 0 6th Sunday since "M" 4:00AM 24 0 0 1 1 0 0 7th Sunday since "M" 4:00AM 25 1 1 1 0 1 1 4th Sunday before "N" after 0:59AM, instead of 26 1 0 1 1 0 1.5t Sunday before "N" after 0:59AM, instead of 30 1 1 1 0 1.5t Sunday of November 0:00AM 0:00AM 31 1 0 0 1 1.5t Sunday of November 1:00AM 0:00AM 32 1 1 1 0 1 3th Sunday of November 1:00AM 33 1 0 1 1 0 1.5t Sunday of November 1:00AM 34 1 0 <t< td=""><td>20</td><td>0</td><td>1</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>4th Sunday of March</td><td>skip from</td></t<>	20	0	1	0	1	1	0	0	4th Sunday of March	skip from				
22 0 0 1 0 0 0 5th Sunday since "M" 4:00AM 23 0 1 1 0 0 1 0 6th Sunday since "M" 24 0 0 1 1 0 0 7th Sunday since "M" 4:00AM 25 1 1 1 0 1 1 4th Sunday before "N" after 0:59AM, instead of 26 1 0 1 0 0 1 2th Sunday before "N" after 0:59AM, instead of 28 1 1 1 1 0 1.5t Sunday before "N" 1:00AM move back to 30 1 1 1 1 0 1.3t Sunday of November 1:00AM 0:00AM 31 1 0 0 1 1.4th Sunday of November 1:00AM 0:00AM 34 1 0 0 1 1.5t Sunday of November 1:00AM 0:00AM 35 1 0 1 0 1.4th Sunday of November 1:00AM 0:00AM 38 <t< td=""><td>21</td><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>4th Sunday since "M"</td><td>3:00AM to</td></t<>	21	0	1	1	1	0	0	0	4th Sunday since "M"	3:00AM to				
23 0 1 1 0 6th Sunday since "M" 24 0 0 1 1 0 0 7th Sunday since "M" 25 1 1 1 1 0 0 7th Sunday since "M" 26 1 0 1 0 1 3th Sunday before "N" after 0:59AM, instead of 27 1 1 1 0 1 1 0 1st Sunday before "N" 28 1 0 1 1 1 1st Sunday of November instead of 30 1 1 1 1 0 1st Sunday of November 0:00AM 31 1 0 0 1 1 0 1st Sunday of November 0:00AM 33 1 0 0 1 1 0 1st Sunday of November after 1:59AM, instead of 38 1 0 0 1 1 1st Sunday of November after 1:59AM, instead of 39 1 0 0 1 1 1st Sunday of November 1:00AM	22	0	0	1	0	0	0	0	5th Sunday since "M"	4.00AM				
24 0 0 1 1 1 0 7th Sunday since "M" 25 1 1 1 1 4th Sunday before "N" after 0:59AM, 26 1 0 1 0 1 3td Sunday before "N" after 0:59AM, 27 1 1 0 1 0 1 3td Sunday before "N" 28 1 0 1 1 0 1st Sunday before "N" after 0:59AM, 29 1 1 0 1 1 0 1st Sunday of November back to 30 1 1 1 1 0 1st Sunday of November back to 31 1 0 0 1 0 1st Sunday before "N" after 1:59AM, 34 1 0 0 1 0 1st Sunday of November after 1:59AM, 35 1 1 0 1 1 1st Sunday of November after 1:59AM, 36 1 0 0 1 1 0std Sunday of November stok to	23	0	1	1	0	0	1	0	6th Sunday since "M"	4.00/101				
25 1 1 1 1 1 4th Sunday before "N" after 0:59AM, instead of 26 1 0 1 0 0 1 3rd Sunday before "N" after 0:59AM, instead of 28 1 0 1 0 1 0 1st Sunday before "N" 1:00AM move 29 1 1 0 1 1 0 1st Sunday of November 30 1 1 1 1 0 1st Sunday of November back to 31 1 1 0 1 1st Sunday before "N" after 1:59AM, instead of 33 1 0 0 1 1st Sunday before "N" after 1:59AM, instead of 35 1 1 0 1 0 1st Sunday before "N" after 1:59AM, instead of 36 1 0 0 1 1 1st Sunday before "N" after 1:59AM, instead of 37 1 0 1 1 0 3rd Sunday of November after 2:59AM, instead of 38 1 0 0 1	24	0	0	1	1	1	0	0	7th Sunday since "M"					
26 1 0 1 0 1 3rd Sunday before "N" after 0:59AM, instead of 27 1 1 1 0 0 1 2nd Sunday before "N" instead of 28 1 0 1 0 1 1 0 1st Sunday before "N" instead of 30 1 1 1 1 0 1st Sunday of November 0:00AM 31 1 1 1 1 0 1st Sunday before "N" after 0:59AM, instead of 32 1 1 1 1 0 1st Sunday of November 0:00AM 33 0 0 1 1 0 1st Sunday before "N" after 1:59AM, instead of 34 1 0 0 1 0 1st Sunday of November after 1:59AM, instead of 35 1 1 0 1 1st Sunday of November after 1:59AM, instead of 38 1 0 0 1 1st Sunday before "N" after 2:59AM, instead of 39 1 0 0 <td< td=""><td>25</td><td>1</td><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td><td>1</td><td>4th Sunday before "N"</td><td></td></td<>	25	1	1	1	0	1	1	1	4th Sunday before "N"					
27 1 1 1 0 0 0 1 2nd Sunday before "N" instead of 28 1 0 1 1 1 0 1st Sunday before "N" instead of 29 1 1 0 0 1 1 0 1st Sunday of November 30 1 1 1 1 1 0 2nd Sunday of November 31 1 1 1 1 0 1 3rd Sunday of November 33 1 0 0 1 0 1 4th Sunday before "N" 34 1 0 0 1 0 1 4th Sunday before "N" 35 1 1 0 1 0 1 0 2nd Sunday of November 38 1 0 0 1 1 1st Sunday before "N" after 1:59AM, instead of 39 1 0 1 1 0 3rd Sunday of November 1:00AM 41 1 1 0 0 1 1 <td< td=""><td>26</td><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td><td>3rd Sunday before "N"</td><td>after 0:59AM,</td></td<>	26	1	0	1	0	1	0	1	3rd Sunday before "N"	after 0:59AM,				
28 1 0 1 0 1st Sunday before "N" 1:00AM move back to 30 1 1 1 1 0 1st Sunday of November 1:00AM move 30 1 1 1 1 0 1:1 1:1 0:0 1:00 Sunday of November 1:00AM 0:00AM 31 1 1 0 0 1:1 0:0 1:1 0:00AM 0:00AM 32 1 1 1 1 0:0 1:1 4th Sunday of November 0:00AM 33 1 0:0 0 1:1 1:0 1:1 4th Sunday of November 1:stead of 35 1 1:0 0:1 1:1 1:1 1:1 1:stead of November 1:stead of 1:00AM 36 1:0 0:0 1:1 1:1 1:1 1:1 1:stead of November 1:00AM 1:00A 1:00A 1:00A <td< td=""><td>27</td><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td><td>2nd Sunday before "N"</td><td>instead of</td></td<>	27	1	1	1	0	0	0	1	2nd Sunday before "N"	instead of				
29 1 1 0 0 1 1 0 1st Sunday of November back to 30 1 1 1 1 1 0 2nd Sunday of November back to 31 1 1 0 0 1 3rd Sunday of November 0.00AM 32 1 1 1 1 0 1 4th Sunday of November 0.00AM 33 1 0 0 1 1 0 1 4th Sunday of November 0.00AM 34 1 0 0 1 0 1 4th Sunday of November after 1:59AM, instead of 35 1 1 0 1 0 2nd Sunday before "N" after 1:59AM, instead of 36 1 0 0 1 1 1 0.2nd Sunday of November after 2:59AM, instead of 39 1 0 0 1 0 3rd Sunday of November after 2:59AM, instead of 44 1 0 1 0 0 2nd Sunday of November after 2:59AM, instead of<	28	1	0	1	0	1	1	0	1st Sunday before "N"	1:00AM move				
30 1 1 1 1 1 1 0 2nd Sunday of November 0:00AM 31 1 1 1 1 0 1 3rd Sunday of November 0:00AM 32 1 1 1 1 0 1 4th Sunday of November 0:00AM 33 1 0 0 1 1 0 1 4th Sunday of November 33 1 0 0 1 1 0 1 4th Sunday of November 34 1 0 0 1 1 0 1 4th Sunday of November 35 1 1 0 1 0 1 1 0 1 0 36 1 0 0 1 1 1 1 1 0 2nd Sunday of November 2:00AM move 38 1 0 0 1 1 0 2nd Sunday of November 1:00AM 41 1 1 0 0 1 0 2nd Sunday of November	29	1	1	0	0	1	1	0	1st Sunday of November	hack to				
31 1 0 0 1 00 1 376 Sunday of November 32 1 1 1 1 0 1 4th Sunday of November 33 1 0 0 1 1 0 1 4th Sunday of November 33 1 0 0 1 1 0 1 4th Sunday of November 34 1 0 0 1 0 1 4th Sunday before "N" 35 1 1 0 1 0 2nd Sunday of November 38 0 0 0 1 1 1st Sunday of November 38 1 0 0 1 1 1st Sunday of November 40 1 1 0 0 1 0 3th Sunday of November 41 1 1 0 0 1 0 3th Sunday of November 44 1 0 1 0 0 1st Sunday of November 45 1 1 0 0 0 <td< td=""><td>30</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>2nd Sunday of November</td><td></td></td<>	30	1	1	1	1	1	1	0	2nd Sunday of November					
33 1 1 1 1 1 1 401 Sunday before "N" 33 1 0 0 1 1 0 1 4th Sunday before "N" 34 1 0 0 0 0 1 4th Sunday before "N" 35 1 1 0 1 0 1 3th Sunday before "N" 36 1 0 0 1 0 1 1 1 1 1 0 1 1 1 0 1 1 1 1 0 1 1 1 0 1 1 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 1 0	31	1	1	0	0	1	0	1	3rd Sunday of November	0:00AM				
33 1 0 0 1	<u>32</u> 22	1	1		1	1	0	1	4th Sunday of November					
35 1 0 0 0 0 1 31d Studay before "N" after 1.35AW, 35 1 0 1 0 1 0 2nd Sunday before "N" instead of 36 1 0 1 0 0 1 1 instead of 37 1 0 1 1 0 1 1 instead of 38 1 0 0 1 1 1 instead of 2:00AM move 39 1 0 0 1 1 0 3rd Sunday of November 1:00AM 40 1 1 0 1 0 3rd Sunday of November 1:00AM 41 1 1 0 0 1 0 3rd Sunday of November 3fter 2:59AM, 43 1 0 1 0 0 3rd Sunday of November 3iter 2:59AM, instead of 44 1 0 1 0 0 3rd Sunday of November 3iter 2:59AM, instead of 47 1	27	 1	0	0	1	1	0	1	3rd Sunday before "N"	after 1.50AM				
36 1 0 1 0 1 0 1	34	1	1	0	1	0	1	0	2nd Sunday before "N"					
37 1 0 1 1 1st Sunday of November 2:00AM move 38 1 0 0 0 1 1 1st Sunday of November back to 39 1 0 0 1 1 1 0 3rd Sunday of November back to 1:00AM 40 1 1 0 0 1 1 0 3rd Sunday of November 1:00AM 41 1 1 0 0 1 0 3rd Sunday of November 1:00AM 41 1 0 1 1 0 0 3rd Sunday of November 1:00AM 43 1 0 0 1 0 0 3rd Sunday before "N" after 2:59AM, instead of 44 1 0 1 0 0 3rd Sunday of November 3:00AM move 45 1 1 1 0 0 3rd Sunday of November 3:00AM move 48 1 0 1 1 0 0 3rd Sunday of November 3:00AM 49<	36	1	0	0	1	0	0	0	1st Sunday before "N"	instead of				
38 1 0 0 0 1 0 2nd Sunday of November back to 39 1 0 0 1 1 0 3rd Sunday of November 1:00AM 40 1 1 0 1 0 3rd Sunday of November 1:00AM 41 1 1 0 0 1 0 4th Sunday of November 1:00AM 42 1 1 0 1 1 0 3rd Sunday before "N" after 2:59AM, 43 1 0 1 0 0 3rd Sunday before "N" after 2:59AM, 44 1 0 1 0 0 3rd Sunday of November after 2:59AM, 45 1 1 0 0 0 1st Sunday of November after 2:59AM, 46 1 1 1 0 0 2nd Sunday of November 3:00AM move 48 1 0 1 1 D 0 3rd Sunday of November 2:00AM 49 x 1 0 1	37	1	0	1	1	0	1	1	1st Sunday of November	2:00AM move				
39 1 0 0 1 1 1 0 3rd Sunday of November 1:00AM 40 1 1 0 1 0 0 1 4th Sunday of November 1:00AM 41 1 1 0 0 1 0 4th Sunday before "N" after 2:59AM, instead of 42 1 0 0 1 0 0 3rd Sunday before "N" after 2:59AM, instead of 43 1 0 0 0 0 1st Sunday before "N" after 2:59AM, instead of 44 1 0 1 0 0 2nd Sunday of November 3:00AM move 45 1 1 0 0 0 3rd Sunday of November 3:00AM 46 1 1 1 0 0 3rd Sunday of November 3:00AM 48 1 0 1 1 0 0 3rd Sunday of November 3:00AM 49 x 1 0 0 1 1 DST in effect for this whole year 51	38	1	0	0	0	0	1	0	2nd Sunday of November	back to				
40 1 1 0 1 4th Sunday of November 41 1 1 0 0 1 0 4th Sunday before "N" 42 1 1 0 1 1 0 3rd Sunday before "N" 43 1 0 0 1 0 0 3rd Sunday before "N" 43 1 0 0 0 1 0 3rd Sunday before "N" 44 1 0 1 0 0 1st Sunday before "N" 45 1 1 0 0 0 1st Sunday of November 46 1 1 1 0 0 2nd Sunday of November 3:00AM move 48 1 0 1 1 0 0 3rd Sunday of November 2:00AM 49 x 1 0 0 1 1 DST transition occurs at different time* 50 x 0 0 1 1 DST in effect for this whole year 51 x 1 0 0 0	39	1	0	0	1	1	1	0	3rd Sunday of November	1:00AM				
41 1 1 1 0 0 1 0 4th Sunday before "N" 42 1 1 0 1 1 0 3rd Sunday before "N" 43 1 0 0 1 0 0 3rd Sunday before "N" 44 1 0 1 0 0 2nd Sunday before "N" after 2:59AM, instead of 44 1 0 1 0 0 0 1st Sunday of November 46 1 1 1 0 0 0 3rd Sunday of November 48 1 0 1 1 0 0 3rd Sunday of November 49 x 1 0 1 1 0 0 3rd Sunday of November 49 x 1 0 0 1 1 DST transition occurs at different time* 50 x 0 0 1 1 1 DST in effect for this whole year 51 x 1 0 0 0 reserved 1 53 <	40	1	1	0	1	0	0	1	4th Sunday of November					
42 1 1 0 1 1 0 0 3rd Sunday before "N" after 2:59AM, instead of 43 1 0 0 0 1 0 0 2nd Sunday before "N" after 2:59AM, instead of 44 1 0 1 0 0 0 1st Sunday before "N" instead of 45 1 1 0 0 0 0 1st Sunday of November back to 46 1 1 1 0 0 0 3rd Sunday of November back to 47 1 1 0 1 0 0 3rd Sunday of November back to 48 1 0 1 1 0 0 3rd Sunday of November back to 49 x 1 0 0 1 1 DST transition occurs at different time* 50 x 0 0 1 1 DST in effect for this whole year 51 x 1 0 0 0 reserved 1 53 x <t< td=""><td>41</td><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td><td>4th Sunday before "N"</td><td></td></t<>	41	1	1	1	0	0	1	0	4th Sunday before "N"					
43 1 0 0 1 0 0 2nd Sunday before "N" instead of 44 1 0 1 0 0 0 1st Sunday before "N" instead of 45 1 1 0 0 0 0 1st Sunday of November instead of 46 1 1 1 0 0 0 2nd Sunday of November back to 47 1 1 0 1 0 0 3rd Sunday of November back to 48 1 0 1 1 0 0 4th Sunday of November 2:00AM 49 x 1 0 0 1 1 DST transition occurs at different time* 50 x 0 0 1 1 1 DST in effect for this whole year 51 x 1 0 0 0 reserved 1 53 x 1 0 0 0 reserved 3 55 x 1 1 0 1 reserved 4	42	1	1	0	1	1	0	0	3rd Sunday before "N"	after 2:59AM,				
44 1 0 1 0 0 0 0 1st Sunday before "N" 45 1 1 0 0 0 0 1st Sunday of November 46 1 1 1 1 0 0 0 2nd Sunday of November back to 47 1 1 1 0 1 0 3rd Sunday of November 2:00AM 48 1 0 1 1 0 0 3rd Sunday of November 2:00AM 49 x 1 0 0 1 1 DST transition occurs at different time* 50 x 0 0 1 1 DST period scheduled this year 51 x 1 0 1 1 DST in effect for this whole year 52 x 1 1 0 0 reserved 1 53 x 1 0 1 0 reserved 3 55 x 1 1 0 1 reserved 4 56 x 1	43	1	0	0	0	1	0	0	2nd Sunday before "N"	instead of				
45 1 1 0 0 0 0 1st Sunday of November 46 1 1 1 1 0 0 0 2nd Sunday of November back to 47 1 1 1 0 1 0 0 3rd Sunday of November back to 48 1 0 1 1 0 0 4th Sunday of November 2:00AM 49 x 1 0 0 1 1 DST transition occurs at different time* 50 x 0 0 1 1 DST in effect for this whole year 51 x 1 0 1 1 DST in effect for this whole year 52 x 1 1 0 0 reserved 1 53 x 1 0 0 reserved 3 55 x 1 1 0 1 reserved 4 56 x 1 1 0 1 reserved 5 x = either 0 or 1, "M"=first Sunday in March, "N"=first Sunday in November ***	44	1	0	1	0	0	0	0	1st Sunday before "N"	3:00AM move				
46 1 1 1 1 0 0 0 2nd Sunday of November 2:00AM 47 1 1 1 0 1 0 0 3rd Sunday of November 2:00AM 48 1 0 1 1 0 0 3rd Sunday of November 2:00AM 48 1 0 1 1 0 0 4th Sunday of November 2:00AM 49 x 1 0 0 1 1 DST transition occurs at different time* 50 x 0 0 1 1 1 DST period scheduled this year 51 x 1 0 1 1 1 DST in effect for this whole year 52 x 1 1 0 0 reserved 1 53 x 1 0 0 reserved 2 54 x 0 1 0 reserved 4 55 x 1 1 0 1 reserved 5 x = either 0 or 1, 1 0	45	1	1	0	0	0	0	0	1st Sunday of November	back to				
47 1 1 1 0 1 0 0 3rd Sunday of November 2:00AM 48 1 0 1 1 0 0 4th Sunday of November 2:00AM 49 x 1 0 0 0 1 1 DST transition occurs at different time* 50 x 0 0 0 1 1 DST transition occurs at different time* 50 x 0 0 1 1 1 DST period scheduled this year 51 x 1 0 1 1 1 DST in effect for this whole year 52 x 1 1 0 0 0 reserved 1 53 x 1 0 0 1 0 reserved 3 55 x 1 1 0 1 reserved 4 56 x 1 1 0 1 reserved 5 x = either 0 or 1, 'M'=first Sunday in March, "N"=first Sunday in November ************************************	46	1	1	1	1	0	0	0	2nd Sunday of November					
48 1 0 1 1 1 0 0 401 Sunday of November 1 49 x 1 0 0 0 1 1 DST transition occurs at different time* 50 x 0 0 0 1 1 1 DST transition occurs at different time* 50 x 0 0 1 1 1 no DST period scheduled this year 51 x 1 0 1 1 1 DST in effect for this whole year 52 x 1 1 0 0 0 reserved 1 53 x 1 0 0 1 0 reserved 2 54 x 0 1 0 1 0 reserved 3 55 x 1 1 0 1 reserved 4 56 x 1 1 0 1 reserved 5 x = either 0 or 1, "M"=first Sunday in March,"N"=first Sunday in November ************************************	47	1	1	1	1	1	0	0	Ath Sunday of November	Z:UUAIVI				
45 x 1 0 0 0 1 1 Destruation occurs at unreferrative 50 x 0 0 0 1 1 1 no DST period scheduled this year 51 x 1 0 1 1 1 DST in effect for this whole year 52 x 1 1 0 0 0 reserved 1 53 x 1 0 0 1 0 reserved 2 54 x 0 1 0 1 0 reserved 3 55 x 1 1 0 1 reserved 4 56 x 1 1 0 1 reserved 5 x = either 0 or 1, "M"=first Sunday in March, "N"=first Sunday in November *Notember *Notember	48		1		1	1	1	1	PST transition occurs at a	lifforont timo*				
S0 x 1 0 1 1 1 10 DST in effect for this whole year 51 x 1 0 1 1 1 DST in effect for this whole year 52 x 1 1 0 0 0 reserved 1 53 x 1 0 0 1 0 0 reserved 2 54 x 0 1 0 1 0 reserved 3 55 x 1 1 0 1 reserved 4 56 x 1 1 0 1 reserved 5 x = either 0 or 1, "M"=first Sunday in March, "N"=first Sunday in November ************************************	50	X	0	0	0	1	1	1	no DST period scheduled	this year				
52 x 1 1 0 0 0 0 reserved 1 53 x 1 0 0 1 0 0 reserved 2 54 x 0 1 0 1 0 0 reserved 3 55 x 1 1 0 1 1 0 reserved 4 56 x 1 1 0 1 reserved 5 reserved 5 x = either 0 or 1, "M"=first Sunday in March, "N"=first Sunday in November *November *November	51	x	1	0	1	1	1	1	DST in effect for this whole	e vear				
53 x 1 0 0 1 0 0 reserved 2 54 x 0 1 0 1 0 0 reserved 2 55 x 1 1 0 1 1 0 reserved 3 56 x 1 1 0 1 1 reserved 5 x = either 0 or 1, "M"=first Sunday in March, "N"=first Sunday in November ************************************	52	x	1	1	0	0	0	0	reserved 1	<u>c ycur</u>				
54 x 0 1 0 1 0 0 reserved 3 55 x 1 1 0 1 1 0 reserved 4 56 x 1 1 0 1 reserved 5 x = either 0 or 1, "M"=first Sunday in March, "N"=first Sunday in November ************************************	53	x	1	0	0	1	0	0	reserved 2					
55 x 1 1 0 1 1 0 reserved 4 56 x 1 1 0 1 0 1 reserved 5 x = either 0 or 1, "M"=first Sunday in March, "N"=first Sunday in November *	54	x	0	1	0	1	0	0	reserved 3					
56 x 1 1 0 1 reserved 5 x = either 0 or 1, "M"=first Sunday in March, "N"=first Sunday in November * DST transition to ensure outside of defined to the def	55	X	1	1	0	1	1	0	reserved 4					
x = either 0 or 1, "M"=first Sunday in March, "N"=first Sunday in November	56	x	1	1	0	1	0	1	reserved 5					
* DCT transition to prove outside of defined ashed also as as a disease with a transition to be	x=eit	her0or1,	"M"=fir	st Sunc	lay in N	larch,	"N"=fir:	st Sund	ay in November					
יכע המוזונטו נס טכנער טענגועפ טו עפווופע schedules, so no advance notification available.	* DST t	ransition t	o occur	outsid	e of def	ined sc	hedule	s, so no	advance notification available					

Table 8 - DST transition schedule word (and reserved messages)

5. Bit Allocation in Message Frame

Table 9 specifies the bit allocation for the message frames. The message frame starts with a 13-bit synchronization word, as defined in subsection 4.2. The LSB of the time word, **time[0]**, is available on bit 19 (marker) in message frames too, as in time frames. This allows receivers that are resolving timing uncertainties only below one minute to be able to use message frames for that purpose, assuming that they are experiencing sufficiently high SNIR for this single 200 ms symbol to suffice.

The **Notice** bit in location 49 (marker), functions in message frames as defined for time frames in subsection 4.5.

The 42 bit word in the remaining locations, {**data[41]**, **data [40]**,...**data[0]**}, defines the contents of the message and may contain fields indicating the address to which the message is intended, the total length of the message (may extend over multiple frames), etc.

Second	0	1	2	3	4	5	6	7	8	9
Legacy AM/PWM	Marker	min_40	min_20	min_10	0	min_8	min_4	min_2	min_1	Marker
Phase	sync_M[12]	sync_M[11]	sync_M[10]	sync_M[9]	sync_M[8]	sync_M[7]	sync_M[6]	sync_M[5]	sync_M[4]	sync_M[3]
Second	10	11	12	12	14	15	16	17	10	10
Jeconu	10		12 h avra 20	13	14	13	10	1/	10	Markar
Legacy AIVI/PWIVI	0	0	nour_20	nour_10	0	nour_8	nour_4	nour_2	1	iviarker
Phase	sync_M[2]	sync_M[1]	sync_M[0]	data[41]	data[40]	data[39]	data[38]	data[37]	data[36]	time[0]
Second	20	21	22	23	24	25	26	27	28	29
Legacy AM/PWM	0	0	day_200	day_100	0	day_80	day_40	day_20	day_10	Marker
Phase	data[35]	data[34]	data[33]	data[32]	data[31]	data[30]	data[29]	data[28]	data[27]	R
Second	30	31	32	33	34	35	36	37	38	39
										Markor
Legacy AM/PWM	day_8	day_4	day_2	day_1	0	0	UT1_S[2]	UT1_S[1]	UT1_S[0]	warker
Legacy AM/PWM Phase	day_8 data[26]	day_4 data[25]	day_2 data[24]	day_1 data[23]	0 data[22]	0 data[21]	UT1_S[2] data[20]	UT1_S[1] data[19]	UT1_S[0] data[18]	R
Legacy AM/PWM Phase	day_8 data[26]	day_4 data[25]	day_2 data[24]	day_1 data[23]	0 data[22]	0 data[21]	UT1_S[2] data[20]	UT1_S[1] data[19]	UT1_S[0] data[18]	R
Legacy AM/PWM Phase Second	day_8 data[26] 40	day_4 data[25] 41	day_2 data[24] 42	day_1 data[23] 43	0 data[22] 44	0 data[21] 45	UT1_S[2] data[20] 46	UT1_S[1] data[19] 47	UT1_S[0] data[18] 48	R 49
Legacy AM/PWM Phase Second Legacy AM/PWM	day_8 data[26] 40 UT1_C_0.8	day_4 data[25] 41 UT1_C_0.4	day_2 data[24] 42 UT1_C_0.2	day_1 data[23] 43 UT1_C_0.1	0 data[22] 44 0	0 data[21] 45 year_80	UT1_S[2] data[20] 46 year_40	UT1_S[1] data[19] 47 year_20	UT1_S[0] data[18] 48 year_10	R 49 Marker
Legacy AM/PWM Phase Second Legacy AM/PWM Phase	day_8 data[26] 40 UT1_C_0.8 data[17]	day_4 data[25] 41 UT1_C_0.4 data[16]	day_2 data[24] 42 UT1_C_0.2 data[15]	day_1 data[23] 43 UT1_C_0.1 data[14]	0 data[22] 44 0 data[13]	0 data[21] 45 year_80 data[12]	UT1_S[2] data[20] 46 year_40 data[11]	UT1_S[1] data[19] 47 year_20 data[10]	UT1_S[0] data[18] 48 year_10 data[9]	R 49 Marker notice
Legacy AM/PWM Phase Second Legacy AM/PWM Phase	day_8 data[26] 40 UT1_C_0.8 data[17]	day_4 data[25] 41 UT1_C_0.4 data[16]	day_2 data[24] 42 UT1_C_0.2 data[15]	day_1 data[23] 43 UT1_C_0.1 data[14]	0 data[22] 44 0 data[13]	0 data[21] 45 year_80 data[12]	UT1_S[2] data[20] 46 year_40 data[11]	UT1_S[1] data[19] 47 year_20 data[10]	UT1_S[0] data[18] 48 year_10 data[9]	Ag Marker notice
Legacy AM/PWM Phase Second Legacy AM/PWM Phase Second	day_8 data[26] 40 UT1_C_0.8 data[17] 50	day_4 data[25] 41 UT1_C_0.4 data[16] 51	day_2 data[24] 42 UT1_C_0.2 data[15] 52	day_1 data[23] 43 UT1_C_0.1 data[14] 53	0 data[22] 44 0 data[13] 54	0 data[21] 45 year_80 data[12] 55	UT1_S[2] data[20] 46 year_40 data[11] 56	UT1_S[1] data[19] 47 year_20 data[10] 57	UT1_S[0] data[18] 48 year_10 data[9] 58	R 49 Marker notice 59
Legacy AM/PWM Phase Second Legacy AM/PWM Phase Second Legacy AM/PWM	day_8 data[26] 40 UT1_C_0.8 data[17] 50 year_8	day_4 data[25] 41 UT1_C_0.4 data[16] 51 year_4	day_2 data[24] 42 UT1_C_0.2 data[15] 52 year_2	day_1 data[23] 43 UT1_C_0.1 data[14] 53 year_1	0 data[22] 44 0 data[13] 54 0	0 data[21] 45 year_80 data[12] 55 LYI	UT1_S[2] data[20] 46 year_40 data[11] 56 LSW	UT1_S[1] data[19] 47 year_20 data[10] 57 DST[1]	UT1_S[0] data[18] 48 year_10 data[9] 58 DST[0]	Agent

Table 9 - Allocation of bits in one-minute message frame - (legacy protocol in black, PM in blue)

6. Example for Transmitted Time Frame

Table 10 clarifies the use of the different fields in the PM protocol through an example, which is explained here. For binary words shown in this example in "{ }", the most-significant bit (MSB) appears to the left and is broadcast first.

The date and time encoded in this example correspond to July 4, 2012 at 17:30 UTC, and would have been broadcast between the instances 17:30:00 and 17:31:00. In other words, the minute being encoded in the broadcast is the one that has already started, as has always been the case with the legacy protocol.

The encoded time is referenced to minute 0, which started at the instance of the turn of the century (00:00:00 UTC on January 1st in the year 2000). Therefore, this example date and time correspond to the 6,578,970th minute of the century, when considering 60 minutes per hour and 24 hours per day. Hence, the 26-bit field **time[25:0]**, corresponding to this instance, holds the binary representation equivalent to the decimal value 6,578,970. Bit **time[25]**, in location 18, represents the MSB and is zero in this example (will not be set to one until after the middle of the century), and bit **time[0]**, in location 46, represents the LSB and is zero in this example since this minute count is an even number. Note that **time[0]** also appears in location 19, which is a marker in the legacy protocol.

Using the parity equations provided in section 4.3, the parity bits for the time-word are found to be {10010}, placed in **time_par[4:0**], where **time_par[4**] is the MSB.

The sync word bits are represented by the 13-bit word $sync_T[12:0]$ (since this is a time information frame rather than a message). These 13 bits are independent of the time and have the fixed values of $\{0011101101000\}$, as specified in Table .3

Since July 4th falls while daylight-saving-time (DST) has been in effect for over one day, the dst_ls[4:0] bits are set to $\{11011\}$ indicating, as shown in Table 4, that the DST state bits dst_on[1:0] are both set to 1, and leap_sec is set to 0. The dst_ls[1:0] bits, being set to $\{11\}$, signify that DST has been in effect for more than one day, and that the next DST transition would be the end of the DST period. The leap_sec notification bit being 0 indicates, as shown in Table 4, that there will be no leap second added to the last minute of this month (July). It is to be noted that the previous month, June 2012, had a leap second added at its end (the minute starting at 23:59:00 UTC on June 30th had 61 seconds), following which the leap second notification bit was to be reset from 1 to 0 until another leap second is decided upon and is to be announced.

The **dst_next[5:0]** field is set to {0 1 1 0 1 1} which, as shown in Table 8, indicates, along with **dst_on[1]** being set to 1, that the transition out of DST is to occur on the first Sunday of November at 02:00AM local time (at the local time zone, if DST is observed, the time is to be moved back to 01:00AM).

The notice bit in location 49, which is a marker, is set to 1, indicating the NIST has a notice posted at <u>http://www.nist.gov/pml/div688/grp40/wwvb.cfm</u>, notifying of changes to the WWVB broadcast signal, anticipated downtime, etc.

In this example, the bits at locations 29 and 39, which are markers, are arbitrarily set to 0 and 1, respectively, carrying no information at this time, as these are reserved for future used.

The bit in location 59, which is a marker, is always set to 0.

The assignment of the AM bits adheres to the legacy protocol, which may be found at this link: <u>http://tf.nist.gov/general/pdf/1383.pdf</u>.

Second	0	1	2	3	4	5	6	7	8	9
Legacy AM/PWM	Marker	min_40	min_20	min_10	0	min_8	min_4	min_2	min_1	Marker
bit value	-	0	1	1	0	0	0	0	0	-
Phase	sync_T[12]	sync_T[11]	sync_T[10]	sync_T[9]	sync_T[8]	sync_T[7]	sync_T[6]	sync_T[5]	sync_T[4]	sync_T[3]
bit value	0	0	1	1	1	0	1	1	0	1
Second	10	11	12	13	14	15	16	17	18	19
Legacy AM/PWM	0	0	hour_20	hour_10	0	hour_8	hour_4	hour_2	hour_1	Marker
bit value	0	0	0	1	0	0	1	1	1	-
Phase	sync_T[2]	sync_T[1]	sync_T[0]	time_par[4]	time_par[3]	time_par[2]	time_par[1]	time_par[0]	time[25]	time[0]
bit value	0	0	0	1	0	0	1	0	0	0
Second	20	21	22	23	24	25	26	27	28	29
Legacy AM/PWM	0	0	day_200	day_100	0	day_80	day_40	day_20	day_10	Marker
bit value	0	0	0	1	0	1	0	0	0	-
Phase	time[24]	time[23]	time[22]	time[21]	time[20]	time[19]	time[18]	time[17]	time[16]	R
bit value	0	0	1	1	0	0	1	0	0	0
Second	30	31	32	33	34	35	36	37	38	39
Legacy AM/PWM	day_8	day_4	day_2	day_1	0	0	UT1_S[2]	UT1_S[1]	UT1_S[0]	Marker
bit value	0	1	1	0	0	0	1	0	1	-
Phase	time[15]	time[14]	time[13]	time[12]	time[11]	time[10]	time[9]	time[8]	time[7]	R
bit value	0	1	1	0	0	0	1	1	0	1
Second	40	41	42	43	44	45	46	47	48	49
Legacy AM/PWM	UT1_0.8	UT1_0.4	UT1_0.2	UT1_0.1	0	year_80	year_40	year_20	year_10	Marker
bit value	0	1	0	0	0	0	0	0	1	-
Phase	time[6]	time[5]	time[4]	time[3]	time[2]	time[1]	time[0]	dst_ls[4]	dst_ls[3]	notice
bit value	0	0	1	1	0	1	0	1	1	1
Second	50	51	52	53	54	55	56	57	58	59
Legacy AM/PWM	year_8	year_4	year_2	year_1	0	LYI	LSW	DST[1]	DST[0]	Marker
bit value	0	0	1	0	0	1	0	1	1	-
Phase	dst_ls[2]	dst_ls[1]	dst_ls[0]	dst_next[5]	dst_next[4]	dst_next[3]	dst_next[2]	dst_next[1]	dst_next[0]	0
hit value	0	1	1	0	1	1	0	1	1	0

Table 10 - Example of one-minute time frame (legacy protocol in black, PM in red)